Demystifying Trajectory Recovery From Ash: An Open-Source Evaluation and Enhancement

Nicholas D'Silva
University of New South Wales
Sydney, Australia
ndsilva64@gmail.com

Toran Shahi
University of New South Wales
Sydney, Australia
toranjungshahi@gmail.com

Øyvind Timian Dokk Husveg
University of New South Wales
Sydney, Australia
timian@husveg.net

Adith Sanjeeve
University of New South Wales
Sydney, Australia
adithsanjeeve1331@gmail.com

Erik Buchholz
University of New South Wales
CSIRO's Data61, Cyber Security CRC
e.buchholz@unsw.edu.au

Salil S. Kanhere
University of New South Wales
Sydney, Australia
salil.kanhere@unsw.edu.au

***Abstract*—Once analysed, location trajectories can provide valuable insights beneficial to various applications, including urban planning, market analysis, and public health surveillance. However, such data is also highly sensitive, rendering them susceptible to privacy risks in the event of mismanagement, for example, revealing an individual's identity, home address, or political affiliations. Hence, ensuring that privacy is preserved for this data is a priority. One commonly taken measure to mitigate this concern is aggregation. Previous work by Xu et al. in [Trajectory Recovery From Ash: User Privacy Is NOT Preserved in Aggregated Mobility Data (2017)] shows that trajectories are still recoverable from anonymised and aggregated datasets. However, the study lacks implementation details, obfuscating the mechanisms of the attack. Additionally, the attack was evaluated on commercial non-public datasets, rendering the results and subsequent claims unverifiable. This study reimplements the trajectory recovery attack from scratch and evaluates it on two open-source datasets, detailing the preprocessing steps and implementation. Results confirm that privacy leakage still exists despite common anonymisation and aggregation methods but also indicate that the initial accuracy claims may have been overly ambitious. We release all code as open-source to ensure the results are entirely reproducible and, therefore, verifiable. Moreover, we propose a stronger attack by designing a series of enhancements to the baseline attack. These enhancements yield higher accuracies by up to 16%, providing an improved benchmark for future research in trajectory recovery methods. Our improvements also enable online execution of the attack, allowing partial attacks on larger datasets previously considered unprocessable, thereby furthering the extent of privacy leakage. The findings emphasise the importance of using strong privacy-preserving mechanisms when releasing aggregated mobility data and not solely relying on aggregation as a means of anonymisation.**

***Index Terms*—Trajectory Recovery, Aggregated Mobility Data, Trajectory Privacy, Location Privacy**

I. Introduction

The domain of human mobility data collection and analysis is of growing importance in both academic and industrial spheres. Some contexts where such data proves beneficial include urban planning, pandemic response analysis, and marketing. Advancements in technology, particularly with personal mobile devices, have facilitated the collection of human mobility data in increasingly higher quantities and quality. However, collecting this data also comes with significant privacy concerns due to the risk of inference of sensitive information. For instance, exploits or mismanagement can lead to identity theft or harassment due to publicised home addresses or personal beliefs. One commonly taken measure to mitigate this concern after anonymisation is aggregation. For example, this may involve transforming a dataset of individual trajectories into a dataset showing the number of individuals within a set of predefined locations over some period of time.

A critical examination of aggregated mobility datasets reveals a serious vulnerability: the ease of re-identification. This concern is highlighted in the work of Xu et al. [1], who demonstrate that aggregated mobility data, despite statistically obfuscating individual records, can be de-anonymised by reconstructing trajectories and potentially revealing sensitive individual information. The authors describe their design as an "elementary but effective attack system to reveal the privacy leakage in aggregated mobility datasets" [1]. They evaluate their attack on two real-world but inaccessible datasets and report accuracies of up to 91 %.

Because of their highly sensitive nature, datasets such as the commercial ones used in the work by Xu et al. are not publicly available, limiting further research possibilities. In conjunction with the fact that the study does not provide any implementation details about their attack, it makes their results irreproducible and renders the claims unverifiable. Intending to increase clarity and transparency in this area, we reimplement the attack they present from scratch, design and implement further enhancements to the attack, and perform evaluations on two public open-source datasets, namely GeoLife [2] and Porto Taxi [3], and release all our code as open-source[1]. Using more accessible datasets allows us to explain with greater transparency the specific characteristics inherent to the aggregated input datasets supposed to contain privacy leakages. We detail

[1] https://github.com/ndsi6382/Trajectory_Recovery

our preprocessing methodology for each of our chosen datasets to ensure our results are reproducible and verifiable. We are convinced that our reimplementation and explanations further clarify the attack process, making it more accessible for further research[2]. Additionally, the enhancements we propose provide a more accurate baseline against which future researchers can benchmark their work, particularly within the field of deep learning. The enhancements also permit the attack to be run online, significantly increasing its accessibility. Partial attacks can be conducted on larger datasets that were previously considered unprocessable by the baseline attack, furthering the extent of the privacy leakage.

Contributions. This work makes the following contributions to the field of location trajectory privacy:

1) Evaluated the validity of the results and claims in [1]:
    a) Reimplemented algorithms from [1].
    b) Preprocessed two publicly available open-source datasets and applied the algorithm to each.
2) Designed, implemented, and evaluated a series of enhancements to the baseline algorithm:
    a) Developed a stronger attack against which future research can use as a baseline.
    b) Showed that our described online methodology allows adversaries to attack larger datasets, furthering the privacy leakage.
3) Encouraged further research, with an emphasis on clarity and transparency:
    a) Released all source code, data, results, and supplementary resources as open-source[1], thus ensuring our results are reproducible and verifiable.
    b) Produced guides detailing our preprocessing and algorithm implementations.
    c) Packaged all algorithms as a Python module with full documentation.

Organisation. In Section II, we contextualise our work, outline a threat model, formally define the problem, summarise the attack from [1], and consequently make clarifying statements about the required properties of the aggregated dataset for the attack to function as intended. We describe our enhancements in Section III. In Section IV, we analyse each open-source dataset and explain our preprocessing methodology. Implementation details are provided in Section V. We evaluate and discuss results in Section VI, mention future directions in Section VII, and provide concluding remarks in Section VIII.

II. Preliminaries

A. Related Work

As the domain of mobility data analysis has grown, so has the emphasis on protecting the privacy of individuals, giving rise to privacy mechanisms [5]–[7] based on k-anonymity [8] and differential privacy (DP) [9]. In recent decades, studies in human mobility have revealed that humans exhibit exceptionally distinctive patterns [10]–[12]. Their trajectories, despite being anonymised, are largely unique and thus pose a risk for re-identification. Consequently, several attacks have been designed targeting anonymised location data, exposing that re-identification is possible with external cross-referenced information [13], and even without [12], [14].

To alleviate concerns of privacy leakage, data collectors and providers often aggregate sensitive information, including locations, before publication. Many recent attacks targeting aggregated location data are membership inference attacks that identify whether an individual's data is included in the supposedly anonymised dataset, giving adversaries access to sensitive information. Examples include the Knock-Knock attack by Pyrgelis et al. [15], with more recent developments by Zhang et al. [16], and Guan and Guépin et al. [17] showing that less or zero prior knowledge is required from adversaries. Other recent developments include individual reconstruction attacks [18], [19] that target reconstructing the original trajectories from a protected (for example, with DP) trajectory dataset. Contrary to these works, the attack evaluated and improved upon in this study reconstructs trajectories from a dataset aggregated by location rather than a protected trajectory dataset. The potential consequences of this are detailed below.

B. Threat Model

The *data owner* is an entity that collects data from *individuals*, such as a mobile phone network provider gathering connection information. The data owner plans to share an aggregated dataset for the *data recipient*'s use. According to the baseline work [1], the data owner is considered trustworthy, and individuals rely on the data owner to adequately anonymise their data (through aggregation). However, neither the data owner nor the individuals trust the data recipient, who acts as an *honest-but-curious adversary* in this threat model. As the baseline work [1], we assume that the data owner is benign and that the individuals trust the data owner to anonymise their data sufficiently (through aggregation). However, there is no trust between either entity and the data recipient, which acts as the *honest-but-curious adversary* in this threat model. The adversary aims to extract as much information as possible about the individuals in the aggregated dataset. The considered attack [1] allows the adversary to recover the trajectories of contained individuals without requiring any background knowledge. While this set of trajectories is still de-identified, existing effective re-identification techniques [12] or trajectory user linking [20], [21] can further be applied, exposing privacy leakage. Note the three-stage design of this attack requires the entire dataset to be processed before re-identification can occur. Therefore, the baseline attack applies only to datasets of a size that can be processed within a realistic timeframe.

In contrast, adversaries can conduct our enhanced attack (detailed in Section III) online, i.e. it processes data and outputs results sequentially without requiring the entire input upfront [22]. Given the same computational resources, this

[2] An alternative implementation [4] of a paper based on [1] exists. However, our version includes the processed open datasets, an abstracted evaluation module, a detailed walk-through of the attack process, and our proposed enhancements, features absent in that existing implementation.

allows the adversary to target subsets of aggregated mobility datasets that were previously considered too large for the baseline attack to process. Noting that re-identification risk only slowly decreases proportionally to the number of individuals in the dataset [23], this method increases the privacy leakage. The modified order of computation allows for data to be processed in chronological order and batches no smaller than one day. Noting that parallel algorithms exist for solving the Linear Sum Assignment problem [24], the remaining computational bottleneck relates to the number of time steps covered by the dataset (see Sections II-D and III). With the online method, intermittent results can be retrieved during execution, allowing contiguous sub-trajectories to be used for re-identification instead of requiring all time steps to be processed first. This yields an increased attack surface.

C. Definitions

Dataset. The baseline attack presented in [1] is formulated as a deterministic algorithm for which we define the problem as follows. An anonymised, aggregated dataset $D \in \mathbb{N}^{t \times m}$ contains t records, where each record $r_i \in \mathbb{N}^{1 \times m}$ for $1 \leq i \leq t$ contains the number of individuals in each of the m locations at the ith time step. Given D, output a set $S \in \mathbb{R}^{n \times t \times 2}$ of reconstructed trajectories for each of the n individuals captured in D, where each trajectory $v_j \in \mathbb{R}^{1 \times t \times 2}$ for $1 \leq j \leq n$ contains the two-dimensional location coordinates for the jth individual, for every time step in chronological order.

Hungarian algorithm. The attack makes extensive use of the Hungarian algorithm (also known as the "Munkres" or "Kuhne-Munkres" algorithm) to solve the square assignment problem [25]. Given a square matrix $C \in \mathbb{R}^{n \times n}$, where rows represent assignees and columns represent assignments, each element $c_{i,j} \in C$ is defined as the cost of assigning i to j. The objective is to determine a one-to-one matching of assignees to assignments that results in the optimal (minimum or maximum) total cost. This is often alternatively described as the Linear Sum Assignment problem [26], where n elements must be selected from the square matrix, subject to the constraints that exactly one element is selected from each row and each column, and to optimise the total sum. The Hungarian algorithm achieves this in $O(n^3)$ time.

D. Attack Summary

The basic mechanism of the baseline attack [1] is to iteratively match each individual's locations of the ith time step with those of the $(i+1)$th. Each assignment produced by the Hungarian algorithm produces the estimated locations for the next time step. The cost matrix at each time step is $C_i \in \mathbb{R}^{n \times n}$, where rows represent individuals, and columns represent locations. While there are actually m locations, the columns enumerate each individual from the aggregated record r_i. For example, if r_i has x many people in location y, then x many columns in C_i shall represent location y. Thus, we must record which columns represent which locations for every time step. Costs are determined by heuristics based on human mobility, primarily leveraging the observation that most people have regular mobility patterns [27], [10].

The first time step of predictions for each day can be trivially determined from the input dataset. Then, the attack is split into three stages. Recovering night-hour trajectories (00:00 to 06:00) is the first stage, where costs are based on physical distance; this is based on the assumption that most people are immobile during night hours. The second stage is recovering the following daytime trajectories (06:00 to 24:00), where the cost is based on a simple velocity model. At the current (ith) time step, given a location p_i and a candidate location ℓ, this heuristic defines the cost of ℓ being the next location as the distance between ℓ and q, where q is the location estimated by extending the vector induced by the locations from the $(i-1)$th and ith time steps:

$$q = p_i + (p_i - p_{i-1}). \tag{1}$$

$$\mathit{cost}(p_i, \ell) = \mathit{distance}(\ell, q). \tag{2}$$

By this point, n sub-trajectories of length d, where d is the number of time steps within a single day, have been recovered for each day captured by the input dataset. For the third stage, each of these sub-trajectories must be uniquely related to each other to recover the full set of n trajectories that last for all t time steps. To obtain this matching, the Hungarian algorithm is again used on a cost matrix where rows represent a day's sub-trajectories and columns represent the next day's sub-trajectories. Based on the observation that people have repetitive daily movements [10], Xu et al. use a standard formulation of information gain to measure the similarity between two sub-trajectories for cost. The three-stage attack is deterministic and runs in $O(tn^3)$ time.

E. Aggregated Dataset Requirements

The baseline attack [1] targets aggregated mobility datasets. Therefore, a suitable dataset must comply with certain requirements. While physical location details are still required, each location must be treated as discrete to represent an area, for example, a mobile base station to which mobile phone users are connected, as per the dataset used in [1]. The aggregated dataset must be complete and contain records of the same set of people, i.e. the total sum within every record must equal n, no records are missing, and no people are unaccounted for. The interval between each time step must be regular and evenly divide 24 hours. The dataset records must begin between 00:00 and 06:00.

III. Design and Heuristic Enhancements

We propose the following design improvements and heuristic enhancements that improve accuracy while reasonably maintaining the efficiency and determinism of the baseline and permit the online execution of the attack.

Stage reduction. The heuristic of the baseline attack assumes trajectories are static during night hours. However, this is not necessarily the case for every person. For example, approximately 18 % of taxi trips from the raw Porto Taxi dataset were recorded as beginning during night hours [3]. For trajectories that do not conform to this assumption, predictions should still accurately consider movement, which, by design, this heuristic does not. Thus, we only use the static distance-based baseline heuristic to generate the location for the time step immediately after the trivial first-step prediction (for midnight) each day. The modified velocity heuristic below determines the remaining $d-2$ predictions for each day. Note that for trajectories that do conform to this static assumption, the velocity heuristic still models immobility accurately and the distribution of locations for the next time step deduced from the input dataset also accounts for this.

Heuristic alterations. We introduce a matrix $B \in \mathbb{N}^{m \times m}$, where each element $b_{i,j} \in B$ is the number of times location j has been predicted to follow location i, for all time steps up to and including the final time step of the last fully-predicted day. This is akin to a bigram or transitional matrix used in algorithms to model hidden Markov processes, such as the Viterbi algorithm [28]. Recording such information allows for future predictions to be affected by historical ones.

The original velocity heuristic is described in Section II-D. Intuitively, this heuristic is restrictive as it does not account for direction changes well, nor does it consider whether the path is common or even possible. For example, if a curved railway surrounded by isolated farmland leads to a popular airport, a linear model may estimate locations leading off the railway and into the farmland. This causes the cost calculation in (2) to be inaccurately based on a poor estimation when an estimation based on popularity was a better choice. Such considerations are similarly helpful for cases where a linear estimation gives a completely inaccessible location, for example, offshore or mountainous. It is, therefore, natural to additionally consider the possible popularity and repetitiveness of certain locations. Inspired by hidden Markov processes, we utilise the information from B by redefining the cost as:

$$H_p = \{i \mid b_{p,i} = \max_{1 \leq j \leq m} (b_{p,j}) \wedge b_{p,i} > 0\}. \tag{3}$$

$$\mathit{cost}(p, \ell) = \min_{x \in H_p \cup \{q\}} \mathit{distance}(\ell, x). \tag{4}$$

where $b_{i,j}$ represents the element in the ith row and jth column of B, and q is defined as per (1).

Sub-trajectory linkage alterations. The baseline attack links sub-trajectories of length d together using information gain as the cost to match similar sub-trajectories. While human mobility patterns are expected to be regular, anomalous trajectories for certain days are still possible. Furthermore, if an incorrect day-to-day linkage is made, this severely impacts the accuracy of the entire trajectory once later linkages are made. Thus, we introduce a positive integer parameter k that expresses the number of previous days to consider. With trajectory u and recently predicted sub-trajectory v, we redefine the cost of linking them as:

$$\mathit{cost}(u, v) = \min_{0 \leq i < k} g(u_{x-i}, v). \tag{5}$$

where u_* means the uth sub-trajectory for the $*$th day, x is the last fully-predicted day, and g is the information gain function, as used in the baseline attack. Note that setting $k = 1$ is equivalent to the linkage mechanism of the baseline attack.

Order of computation. The design of the baseline attack considers these linkages as the "third stage", to be performed after all sub-trajectories of length d are independently recovered. Our alterations require that the trajectories be recovered chronologically and linked cumulatively. After each sub-trajectory of length d is predicted, they must be linked to the existing trajectories, followed by an update to the bigram matrix B. This modified order of computation enables the algorithm to be run online, the ramifications of which are outlined in Section II-B.

With these alterations, the algorithm remains deterministic and requires minimal additional computational resources. During experimentation, it was determined that values for $k > 7$ (representing a repetition schedule of more than one week) were of little to no benefit to the accuracy, so we reasonably assume that $k \ll m$ and $k \ll n$. Then, the enhanced attack runs in $O(t(n^3 + n^2 m))$ time and requires additional $O(m^2)$ space compared to the baseline attack. Note that whether $m > n$ depends on external factors of the input dataset, such as population density and spatial resolution. The results of these enhancements are presented in Section VI.

IV. Data

A. Dataset 1: GeoLife

GeoLife [2], collected by Microsoft Research Asia, is a dataset of time-stamped GPS locations from 182 different users in Beijing, China. The data spans from 2007 to 2012, but most users are inactive for most of this period. It contains over 17 000 trajectories with a total time of over 48 000 hours. As mentioned in Section II-E, we require a dataset with uniform time intervals, however the raw dataset has variable time intervals. Furthermore, it is sparse, with too few users with trajectories in a common period. We apply the following manipulations to address these issues.

Most spatiotemporal points are located within a certain region of Beijing. Points outside Beijing and users with only a single trajectory are discarded. Then, only records from the top-k most active months for each user are retained. To partially address the inconsistent interval of time stamps, a floor operation is applied to each time stamp, followed by removing duplicates caused by this operation. The remaining interval-related inconsistencies are resolved with interpolation (see Section IV-C).

Additionally, some users have multiple trajectories that span only a few hours, complicating finding a common period with multiple active users. To resolve this, we shift consecutive trajectories temporally closer to each other. If these are also deemed to be reasonably close spatially, they are merged to create a longer trajectory. As one consistent trajectory is required, only the longest trajectory for each user is kept. Then, we assign the same start date-timestamp to all trajectories and adjust the trajectories to follow the new starting date. With all the trajectories aligned, all points beyond the earliest ending time-stamp are discarded, resulting in a dataset temporal coverage of approximately one week.

B. Dataset 2: Porto Taxi

Porto Taxi [3] is a dataset of spatiotemporal points collected from taxis in Porto, Portugal. It contains 1 710 670 trajectory records of 444 taxis over the span of one year, from 01/07/2013 to 30/06/2014. Each trajectory represents a taxi trip and is given as a list of GPS coordinates captured at an interval of approximately 15 s. Some trajectories have missing points; these are immediately discarded, leaving 1 704 685 trajectories.

In addition to complying with the requirements in Section II-E, the objective is to retain a maximal dataset size while minimising our interference. To achieve this, we identify the densest period of taxi trips in the year, then retain only the trajectories of taxis that have completed an above-average number of trips within that period, thus limiting the application of interpolation described in Section IV-C. The monthly maximum number of taxi trips is in May, the first full week of which has the weekly maximum. After filtering as described, 197 trajectories remain. Then, each trajectory is sampled every 10 min, resulting in 4321 time steps for a 30 day period.

C. General Preprocessing Steps

After completing the specific preprocessing mentioned above, the following operations are required.

No location records are available for the period between two records where users are idle. This gap is addressed by filling it with interpolated records, though we must note that no interpolation technique can entirely reflect a true mobility pattern. A static interpolation method was deemed the most appropriate, where the user's last-known location is repeated for each time step until the next-known location. This was selected over other techniques, such as linear interpolation, where missing locations are filled by regularly spacing locations between the last-known and next-known locations for each intermediate time step because such a method forces trajectories to contain locations that may not exist or be accessible.

To enforce the locations as discrete areas, as shown in Fig. 1, a rectangular region is defined with the bottom-left and top-right corners at the 1st and 99.5th percentiles of latitude and longitude, respectively. Any spatiotemporal points outside this enforced boundary are shifted to the boundary, ensuring that the true location is represented in at least one spatial dimension. The region is then divided into square cells; points located within a cell are said to belong to that 'location'. The cells were set to represent an area of $1\,\mathrm{km}^2$ for GeoLife, and $4\,\mathrm{km}^2$ for Porto Taxi, reflecting practical spacings between modern cellular network towers in urban areas [29]. The physical location is considered to be the centre of that cell. A user's location at any time step is taken as the ground truth location within each trajectory. The total number of users in each location at each time step is taken as aggregated data.

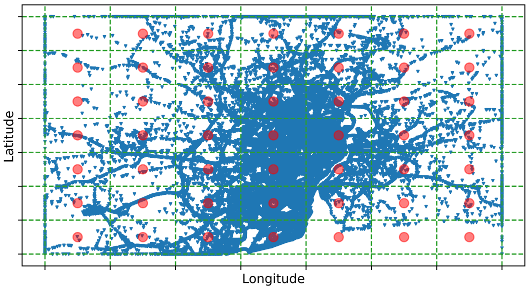

Fig. 1. Location grid with cell centres (red) and user locations (blue) applied to the Porto Taxi dataset.

D. Limitations

As outlined, extensive preprocessing was necessary to align the datasets with the attack framework's requirements (see Section II-E). We could not obtain the datasets used in the original publication due to access restrictions, and we are unaware of any other high-quality open-source datasets with similar properties that could have been used instead. Additionally, the Porto Taxi dataset specifically targets taxi drivers, whose mobility patterns differ significantly from typical mobile phone users. After preprocessing, the number of users in both datasets was considerably lower than in the original study. These constraints have inevitably impacted our results, but there were no alternatives for conducting the attack using openly available data.

V. Implementation Details

All code is provided in Python 3.10 and released as open-source under the MIT licence[1]. The implementation deploys the Numpy, Pandas, Matplotlib, Scipy, Geopy, Levenshtein, and Tqdm packages. All preprocessing code is released as annotated Jupyter Notebooks describing the process in detail. The baseline attack [1] was re-implemented in a Jupyter Notebook that contains extensive explanations detailing each step of the algorithm and shows intermediate results. This representation clarifies the nature of the privacy leakage and details the attack mechanisms and the features of the data that lead to the leakage. Moreover, we provide Python classes for both the baseline and enhanced attacks that allow execution via scripts. The modular nature of this implementation allows for (additional) datasets to be readily loaded, visualised, and evaluated with minimal additional code, facilitating further research. The repository further contains full API documentation. The implementation of our enhanced attack allows predictions to be accessed online from another thread during execution.

TABLE I
Sub-dataset Details

Dataset	#Users	#Locations	#Time Steps	Interval
GeoLife 37	37	194	5205	2 min
GeoLife 38	38	492	10 407	1 min
GeoLife 43	43	120	10 103	1 min
Porto Taxi 3×3	197	9	4321	10 min
Porto Taxi 7×7	197	49	4321	10 min

VI. Evaluation and Discussion

Metrics. To draw comparisons, the metrics used for evaluation mirror those utilised by Xu et al. [1]. In their study, they define *accuracy* as the proportion of correctly predicted spatiotemporal points, given by:

$$accuracy = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i \cap B_i|}{t}. \qquad (6)$$

where A_i and B_i represent the ith predicted and corresponding true trajectories respectively. This can be described as the average of the complement of the normalised Hamming distances [30] between every predicted trajectory and the associated (see Mapping below) true trajectory. Additionally, they define the *recovery error* as the total sum of distances between predicted and true spatiotemporal points. They also introduce the *top-k uniqueness* [12] of a dataset as "the percentage of recovered trajectories that can be uniquely distinguished by their most frequent k locations" [1]. This metric quantifies how easily the recovered trajectories can re-identify individuals, completing the de-anonymisation process. A natural example is $k = 2$, where the two most frequent locations can be assumed to be one's home and workplace [31]. Intuitively, high uniqueness indicates more severe levels of privacy leakage in the dataset, as individuals are more unique regarding the places they frequent and are, therefore, easier to identify.

Mapping. To apply these metrics, the predicted trajectories must be associated with true trajectories by creating a one-to-one matching between the two sets. To achieve this, we create another cost matrix $C \in \mathbb{R}^{n\times n}$, where costs are defined as the recovery error between the two trajectories, and apply the Hungarian algorithm to produce this mapping. By contrast, the mapping method used in [1] is greedy, achieved by iteratively matching each predicted trajectory with the most similar unmatched true trajectory. This potentially results in sub-optimal pairings, which negatively affects the accuracy. We opted for the Hungarian algorithm, as it ensures that the recovery error is globally minimised, maximising the accuracy. Note that the improved mapping was used for both baseline and enhanced attacks to ensure a fair comparison.

Following the preprocessing outlined in Section IV, we ultimately obtained three sub-datasets from GeoLife and two from Porto Taxi. We evaluated the baseline and the enhanced attack on each of these. The details of each sub-dataset are shown in Table I. The accuracies, recovery errors, and top-k uniquenesses are shown in Figs. 2, 4, and 3, respectively.

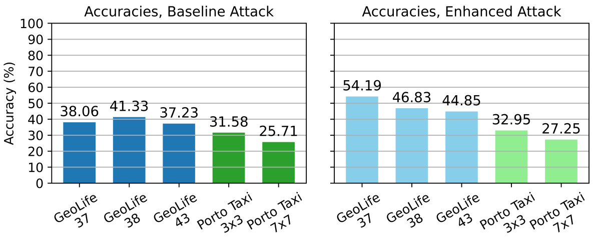

Fig. 2. Accuracies on the baseline (left) and enhanced (right) attacks.

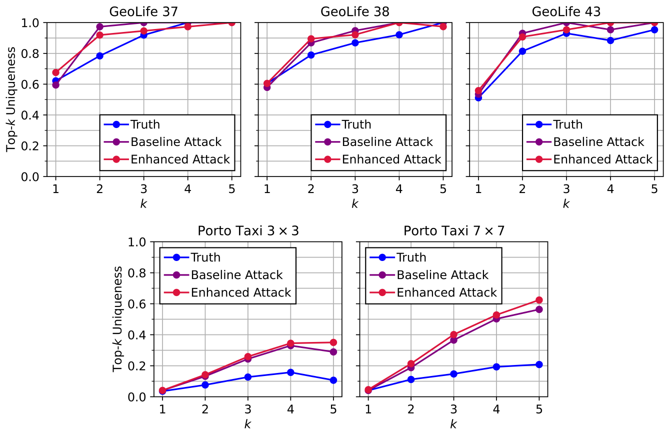

Fig. 3. Top-k uniqueness values for $1 \leq k \leq 5$. The ground truth and outputs of the baseline and enhanced attacks are shown for each sub-dataset.

As shown in Fig. 2, the attack was more successful on the GeoLife datasets. The highest accuracy achieved by the baseline attack was $41\,\%$ on the 38-user dataset, with our enhancements increasing this to $46\,\%$. The enhanced attack's highest accuracy was achieved on the 37-user dataset, reaching $54\,\%$, while the baseline algorithm achieved $38\,\%$, highlighting the largest marginal improvement from our enhancements. More generally, the enhanced version yielded higher accuracies for every sub-dataset across both datasets, especially for those derived from GeoLife. All measurements for our enhancement were conducted with a linkage parameter of $k = 3$.

Overall, these accuracies are significantly lower than those achieved by Xu et al. on their commercial datasets, where they achieved 73-91 % using the baseline attack. This is partially expected given our explanations of the limitations of our datasets in Section IV-D. We expect that with our much smaller datasets, the accuracy metrics suffer from granularity-related noise. We also note that GeoLife is a human mobility dataset, while Porto Taxi is a vehicular mobility dataset, and that the heuristics used in [1] that we replicate and extend are based primarily on human mobility. This explains the lower accuracies in general from the Porto Taxi dataset and the marginally smaller improvements from the enhancements. Given that the heuristics designed by Xu et al. were based on human mobility patterns, we hypothesise that using heuristics tailored to other types of mobility should give more accurate results for those kinds of data. Fig. 3 also shows that uniqueness values for GeoLife are far higher, facilitating higher accuracies. The vehicular nature of the Porto Taxi dataset may also explain its low uniqueness values. For example, popular points of interest (such as an airport) and routes (such as arterial highways)

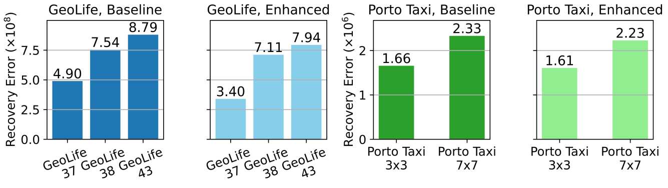

Fig. 4. Recovery errors on the baseline and enhanced attacks.

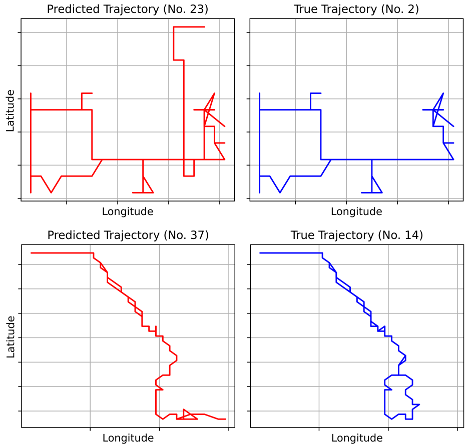

Fig. 5. Two examples of predicted trajectories and their matched true trajectories over a single day.

reduce the uniqueness of the data, resulting in less accurate trajectories, as shown by the results. However, the results of the GeoLife dataset suggest that our enhancements can recover trajectories more accurately than the original attack.

We also observe the trajectories predicted by the attack compared to their associated true trajectories in Fig. 5. In both examples, the trajectories are mostly recovered, apart from some outlying patterns. In situations where the locations are not exactly correct, we also observe that the heuristics described in [1] must somewhat accurately capture patterns in human mobility. Hence, we conclude that there is some privacy leakage from the processed aggregated datasets.

A downside of using the Hamming distance in an accuracy measure is the possibility of a trajectory matching attaining poor accuracy due to minor errors in one of the spatiotemporal dimensions. An example of such a situation occurs when a trajectory contains an almost perfect sub-sequence of locations but is incorrectly shifted one time step. In terms of edit distance operations, the Hamming distance only permits substitutions [32]. To additionally consider insertions and deletions, we evaluated the Levenshtein distance [33] and used it as an alternative measure of accuracy as follows:

$$\mathit{Levenshtein\ accuracy} = \frac{1}{n}\sum_{i=1}^{n} 1 - \frac{L(A_i, B_i)}{t}. \qquad (7)$$

where L represents the Levenshtein distance function.

Fig. 6 shows the Levenshtein metrics on each sub-dataset with both attack versions. The results each evaluate slightly higher than the accuracies shown in Fig. 2, but generally follow the same profile, further confirming the reliability of these metrics on these datasets.

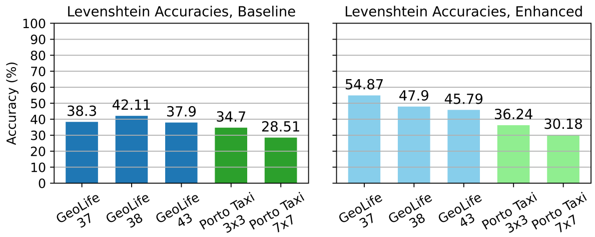

Fig. 6. Levenshtein accuracies on baseline (left) and enhanced (right) attacks.

Although these results show that the recovery of trajectories is possible with the baseline attack, they suggest that initial claims about the accuracy and, therefore, the severity of privacy leakage may have been overly ambitious. Despite GeoLife being a human mobility dataset and preprocessed in such a way as to mimic the commercial mobile operator dataset used in [1] as closely as possible, there is a significant disparity between the results, even when including our enhancements. This demonstrates a lack of ability for the attack to generalise to other datasets and highlights that the specific datasets used in [1] are exceptionally well-suited for the task. Based on our evaluations, we claim that accuracies of up to 91 % somewhat misrepresent the capabilities of such an attack in real-world scenarios. The baseline attack's outstanding performance in [1] leans on strong assumptions for the considered dataset and the contained users. The inaccessible datasets used by the baseline contained many trajectories with fine-granular sampling over long periods of time for a fixed set of users. Moreover, the users seemed to comply with the strong assumptions about human mobility made by the authors, such as very limited movement during night hours. While reporting leakage in a worst-case setting is important, such assumptions do not transfer to real-world datasets that commonly suffer from less regular and uniform samples, such as the considered GeoLife and Porto Taxi datasets. Thus, the remarkably high accuracies exceeding 90 % in the baseline work [1] might significantly overestimate the success of the attack in practical scenarios. Nevertheless, the results confirm that privacy leakage is a real concern, and the risk should not be underestimated.

VII. Future Work

Throughout previous sections, we established how heuristics significantly influence the performance of this attack. Naturally, it is interesting to investigate whether using heuristics tailored to other forms of mobility results in similar privacy leakages for other types of mobility data. Further experimentation on more human mobility datasets (as they become available) will also clarify the disparity between the results achieved on our public open-source datasets and the private commercial ones evaluated by Xu et al. [1].

The heuristic methods used by us and in [1] are limited in their ability to fully encapsulate the dynamic nature of human movements because they are based on researcher-defined assumptions that potentially overgeneralise the more complex patterns evident in human mobility [10]. Prior research in location trajectory privacy suggests that deep-learning methods may be able to address this. For example, Wang et al. [34] explore the usage of LSTMs and Seq2Seq approaches for the trajectory prediction task, and multiple authors [20], [21] leverage Recurrent Neural Networks for trajectory user linking. Their results illustrate the potential of deep learning to improve accuracy and robustness over static heuristic methods.

VIII. Conclusion

To evaluate the privacy leakage of aggregated mobility datasets, we successfully reimplemented the trajectory recovery attack proposed by Xu et al. [1]. The original study evaluated the attack on inaccessible commercial datasets, rendering the results irreplicable and the subsequent claims unverifiable. To increase transparency in this area of research, we initially conducted the same attack with our reimplementation, using public open-source datasets, namely GeoLife [2] and Porto Taxi [3]. To further facilitate future research, we designed improvements to the baseline that yielded substantially higher accuracies (by up to $16\,\%$), requiring minimal additional computation, for use as an improved baseline. Our improvements also permit an online version of the attack, making the attack significantly more accessible to larger datasets previously considered unprocessable. We released all code as open-source to ensure our findings are reproducible. Our results, attaining accuracies of up to $54\,\%$ on the GeoLife dataset and $32\,\%$ on the Porto Taxi dataset, show that the reconstruction of individual trajectories from anonymised aggregated data represents a practical risk. The results confirm the privacy concerns raised by Xu et al. but also suggest that the originally reported results are over-exaggerated and depend on strong assumptions about the considered dataset and users. Nevertheless, this work emphasises the need for enhanced privacy protection measures when publishing aggregated mobility data.

Acknowledgment

The authors would like to thank UNSW, the Commonwealth of Australia, and the Cybersecurity Cooperative Research Centre Limited for their support of this work.

References

[1] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, “Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data,” in *Proc. 26th Int. Conf. World Wide Web*, WWW ’17, Int. World Wide Web Conf. Steering Committee, Apr. 2017.

[2] Y. Zheng, H. Fu, X. Xie, W.-Y. Ma, and Q. Li, *Geolife GPS trajectory dataset - User Guide*, geolife gps trajectories 1.1 ed., July 2011.

[3] M. O’Connell, L. Moreira-Matias, and W. Kan, “Ecml/pkdd 15: Taxi trajectory prediction (i),” 2015.

[4] Z. Tu, “Trajectory recovery.” https://github.com/tuzhen8000/Trajectory_Recovery, 2018.

[5] V. Primault, A. Boutet, S. B. Mokhtar, and L. Brunie, “The long road to computational location privacy: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2772–2793, 2019.

[6] A. Miranda-Pascual, P. Guerra-Balboa, J. Parra-Arnau, J. Forné, and T. Strufe, “SoK: Differentially Private Publication of Trajectory Data,” *Proc. Priv. Enhancing Technol. (PoPETs)*, vol. 2023, no. 2, 2023.

[7] E. Buchholz, A. Abuadbba, S. Wang, S. Nepal, and S. S. Kanhere, “SoK: Can Trajectory Generation Combine Privacy and Utility?,” *Proc. Priv. Enhancing Technol. (PoPETs)*, vol. 2024, July 2024.

[8] L. Sweeney, “k-anonymity: a model for protecting privacy,” *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, oct 2002.

[9] C. Dwork, “Differential privacy,” in *Automata, Languages and Programming* (M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, eds.), (Berlin, Heidelberg), pp. 1–12, Springer Berlin Heidelberg, 2006.

[10] M. C. González, C. A. Hidalgo, and A.-L. Barabási, “Understanding individual human mobility patterns,” *Nature*, vol. 453, June 2008.

[11] K. Sui, Y. Zhao, D. Liu, M. Ma, L. Xu, L. Zimu, and D. Pei, “Your trajectory privacy can be breached even if you walk in groups,” in *2016 IEEE/ACM 24th Int. Symposium on Quality of Service (IWQoS)*, pp. 1–6, 2016.

[12] Y.-A. Montjoye, C. Hidalgo, M. Verleysen, and V. Blondel, “Unique in the crowd: The privacy bounds of human mobility,” *Scientific reports*, vol. 3, p. 1376, 03 2013.

[13] R. Shokri, G. Theodorakopoulos, G. Danezis, J.-P. Hubaux, and J.-Y. Le Boudec, “Quantifying location privacy: The case of sporadic location exposure,” in *Privacy Enhancing Technologies* (S. Fischer-Hübner and N. Hopper, eds.), (Berlin, Heidelberg), pp. 57–76, Springer Berlin Heidelberg, 2011.

[14] S. Gambs, M.-O. Killijian, and M. Núñez del Prado Cortez, “De-anonymization attack on geolocated data,” *Journal of Computer and System Sciences*, vol. 80, no. 8, pp. 1597–1614, 2014.

[15] A. Pyrgelis, C. Troncoso, and E. D. Cristofaro, “Knock knock, who’s there? membership inference on aggregate location data,” *CoRR*, vol. abs/1708.06145, 2017.

[16] G. Zhang, A. Zhang, and P. Zhao, “Locmia: Membership inference attacks against aggregated location data,” *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11778–11788, 2020.

[17] V. Guan, F. Guépin, A.-M. Cretu, and Y.-A. de Montjoye, “A zero auxiliary knowledge membership inference attack on aggregate location data,” *Proc. Priv. Enhancing Technol. (PoPETs)*, vol. 2024, no. 4, p. 80–101, 2024.

[18] M. Shao, J. Li, Q. Yan, F. Chen, H. Huang, and X. Chen, “Structured Sparsity Model Based Trajectory Tracking Using Private Location Data Release,” *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 6, pp. 2983–2995, 2020.

[19] E. Buchholz, A. Abuadbba, S. Wang, S. Nepal, and S. S. Kanhere, “Reconstruction Attack on Differential Private Trajectory Protection Mechanisms,” in *Proc. 38th Annu. Comput. Secur. Appl. Conf.*, ACSAC ’22, (New York, NY, USA), pp. 279–292, Association for Computing Machinery, Dec. 2022.

[20] Q. Gao, F. Zhou, K. Zhang, G. Trajcevski, X. Luo, and F. Zhang, “Identifying human mobility via trajectory embeddings,” *Proc. 26th Int. Joint Conf. on Artificial Intelligence*, Aug 2017.

[21] L. May Petry, C. Leite Da Silva, A. Esuli, C. Renso, and V. Bogorny, “MARC: A robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings,” *Int. J. Geogr. Inf. Sci.*, vol. 34, no. 7, pp. 1428–1450, 2020.

[22] R. M. Karp, “On-line algorithms versus off-line algorithms: How much is it worth to know the future?,” in *IFIP Congress*, 1992.

[23] A. Farzanehfar, F. Houssiau, and Y.-A. de Montjoye, “The risk of re-identification remains high even in country-scale location datasets,” *Patterns*, vol. 2, no. 3, p. 100204, 2021.

[24] K. Date and R. Nagi, “GPU-accelerated Hungarian algorithms for the Linear Assignment Problem,” *Parallel Comput.*, vol. 57, 2016.

[25] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[26] G. B. Dantzig, *Linear Programming and Extensions*. Santa Monica, CA: RAND Corporation, 1963.

[27] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, “Limits of Predictability in Human Mobility,” *Science*, vol. 327, no. 5968, 2010.

[28] G. D. Forney, “The viterbi algorithm,” *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[29] B. Prkić, “Understanding small-cell wireless backhaul,” Apr 2014.

[30] R. W. Hamming, “Error detecting and error correcting codes,” *The Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[31] P. Golle and K. Partridge, “On the anonymity of home/work location pairs,” in *Pervasive Computing* (H. Tokuda, M. Beigl, A. Friday, A. J. B. Brush, and Y. Tobe, eds.), (Berlin, Heidelberg), pp. 390–397, Springer Berlin Heidelberg, 2009.

[32] G. Navarro, “A guided tour to approximate string matching,” *ACM Comput. Surv.*, vol. 33, p. 31–88, Mar 2001.

[33] V. I. Levenshtein, “Binary Codes Capable of Correcting Deletions, Insertions and Reversals,” in *Sov. Phys. Dokl.*, vol. 10, p. 707, 1966.

[34] C. Wang, L. Ma, R. Li, T. S. Durrani, and H. Zhang, “Exploring trajectory prediction through machine learning methods,” *IEEE Access*, vol. 7, p. 101441–101452, Jul 2019.