Machine Learning Aided Anonymization of Spatiotemporal Trajectory Datasets

Sina Shaham*, Ming Ding**, Bo Liu^{+}, Zihuai Lin*, Jun Li^{++}

*School of Electrical and Information Engineering, The University of Sydney, Australia
**Data61, CSIRO, Sydney, Australia
$^{+}$Department of Engineering, La Trobe University, Australia
$^{++}$Nanjing University of Science and Technology, Nanjing, China
Email: {sina.shaham, zihuai.lin}@sydney.edu.au, ming.ding@data61.csiro.au, b.liu2@latrobe.edu.au, jun.li@njust.edu.cn

Abstract—**The big data era requires a growing number of companies to publish their data publicly. Preserving the privacy of users while publishing these data has become a critical problem. One of the most sensitive sources of data is spatiotemporal trajectory datasets. Such datasets are extremely sensitive as users' personal information such as home address, workplace and shopping habits can be inferred from them. In this paper, we propose an approach for anonymization of spatiotemporal trajectory datasets. The proposed approach is based on generalization entailing alignment and clustering of trajectories. We propose to apply k'-means algorithm for clustering trajectories by developing a technique that makes it possible. We also significantly reduce the information loss during the alignment by incorporating multiple sequence alignment instead of pairwise sequence alignment used in the literature. We analyze the performance of our proposed approach by applying it to Geolife dataset, which includes GPS logs of over 180 users in Beijing, China. Our experiments indicate the robustness of our framework compared to prior works.**

I. Introduction

Publishing data by different organizations and institutes is crucial for open research and transparency of government agencies. Just in Australia, since 2013, over 7000 additional datasets have been published on 'data.gov.au', a dedicated website for data publication by the Australian government. Moreover, the new Australian government data sharing and legislation [1]–[3] encourages government agencies to publish their data, and as early as 2018 many of them will have to do so. The process of data publication can be significantly risky as it may contain individuals' sensitive information. Therefore, an essential step before publishing data is to remove any personally identifying information from the dataset. However, such operation is not sufficient for privacy preservation. Adversaries are able to link the datasets using common attributes called quasi-identifiers, or may have prior knowledge about the trajectories travelled by the users which enables them to reveal sensitive information that can cause physical, financial and reputational harms to people.

One of the most sensitive sources of data is location trajectories or spatiotemporal trajectories. Despite numerous use cases that the publication of spatiotemporal data can provide to users and researchers, it poses a significant threat to users' privacy. As an example, consider a person who has been using GPS navigation to travel from home to work every morning of weekdays. If an adversary has some prior knowledge about a user, such as the home address, it may be able to identify the user. This can compromise private information about the user, such as the user's health condition and how often does the user visit his/her specialist. Therefore, it is crucial to anonymize spatiotemporal datasets before publishing them to the public.

Most of the work in the area of privacy preservation for spatiotemporal trajectories is focused on achieving k-anonymity proposed in [4]. The idea is to hide the true data among at least $k-1$ other data entries so that the trajectories of the users are not distinguishable. The authors in [5], adopted the notion of k-anonymity for trajectories and proposed an anonymization algorithm based on generalization. Xu et al. [6] investigated the factors such as spatio-temporal resolution and the number of users released. The authors in [7] focused on improving the clustering approach in the anonymization process. The proposed anonymization scheme is based on achieving k-anonymity by grouping similar trajectories and removing the ones that are highly dissimilar. More recently, the authors in [8] developed an algorithm called k-merge to anonymize the trajectory datasets while preserving the privacy of users from probabilistic attacks. Local suppression and splitting techniques were considered in [9].

In general, there are two major problems with the existing methods. Firstly, it lacks well-defined methods to cluster trajectories as there is no easy way to measure the cost of clustering according to distances among trajectories. Secondly, the existing literature focuses on pairwise sequence alignment which results in a high amount of information loss during the trajectory alignment. **In our work, we address these problems by proposing a method to anonymize spatiotemporal trajectories. Our approach has two main contributions:**

- **Use of multiple sequence alignment instead of pairwise sequence alignment, which significantly reduces the cost of alignment.**
- **Applying k'-means clustering algorithm for clustering trajectories by developing a technique to enable the approach.**

We analyze the performance of our proposed anonymization method by applying it to Geolife dataset which includes GPS logs of over 180 users in Beijing, China. Our experiments indicate the robustness of our proposed method compared to recent work in [8].

The rest of this paper is organized as follows. We start by explaining the system model and formulating the problem in section II. The proposed anonymization approach is presented in section III, followed by experimental results in section IV. Finally, we conclude the paper in section V.

II. System Model & problem formulation

We assume that a map has been discretized into an $\epsilon \times \epsilon$ grid and the time is discretized into bins of length ϵ_t. Therefore, each point in the dataset represents a snapshot of a real-world location query including x-coordinate, y-coordinate, and time. In our model, we consider a spatiotemporal trajectory datasets denoted by T. The dataset consists of trajectories $tr_1, ..., tr_n$ where n represents the number of trajectories in the dataset ($T = \{tr_1, ..., tr_n\}$, $|T| = n$). Each trajectory tr_i is an ordered set of l_i spatiotemporal 3D points ($tr_i = \{p_1, ..., p_{l_i}\}$, $|tr_i| = l_i$). Each point p_j is a triplet of x-coordinate, y-coordinate, and the time of query, respectively.

A. Privacy and Threat Models

In this paper, we adopt a well-known metric called k-anonymity [4] to ensure the privacy of users.

Definition 1. *k-anonymous dataset:* A trajectory dataset $\overline{T}$ is a k-anonymization of a trajectory dataset T if for every trajectory in the anonymized dataset $\overline{T}$, there are at least $k-1$ other trajectories with exactly the same set of points, and there is a one to one mapping relationship between the trajectories in $\overline{T}$ and T.

For the threat model, we assume that no uniquely identifiable information is released while publishing the dataset. However, the adversary may:

- already know about part of the released trajectory for an individual and attempt to identify the rest of the trajectory.
- already know the whole trajectory that an individual has travelled, but try to access other information released while publishing the dataset by identifying the user in the dataset.

Our aim is to protect the users against the adversary's attempt to access sensitive information that may compromise the privacy of users.

B. Hierarchical Tree Transformation

To anonymize the dataset, generalization and suppression techniques are used based on domain generalization hierarchy (DGH). A DGH for attribute $\mathcal{A}$, referred to as $H_{\mathcal{A}}$, is a partially ordered tree structure which maps specific and generalized values of the attribute $\mathcal{A}$. The root of the tree is the most generalized value and is returned by function RT which contains zero bits of information.

Example 1. Consider an 8×8 map. The x-coordinate attribute can have 8 possible values $(0, 1, ..., 7)$. DGH divides the largest possible interval for x-coordinate ($[0-7]$), which is the root of the tree, to two, four, and eight x-coordinate intervals as DGH increases in depth. In this example, the lowest level of the tree can be shown by 3 bits. One bit of information loss incurs by moving up one level in the tree. Fig. 1 shows the structure of x-coordinate DGH. For the generation of y-coordinate and time DGHs, a similar approach can be taken.

Each node on a DGH can be generalized by moving up one or multiple levels of the DGH. The process of generalizing node$_i$ to one of its parent nodes node$_j$ is denoted using node$_i$ $\rightarrow$ node$_j$. A special case of generalization, in which the node is generalized to the root of the DGH, is referred to as suppression. These two techniques are used as tools to anonymize the dataset in the following sections. It must be noted that although quasi-identifiers in this paper are x-coordinate, y-coordinate, and the time of query, the algorithms developed in our work can be extended to include other attributes as well.

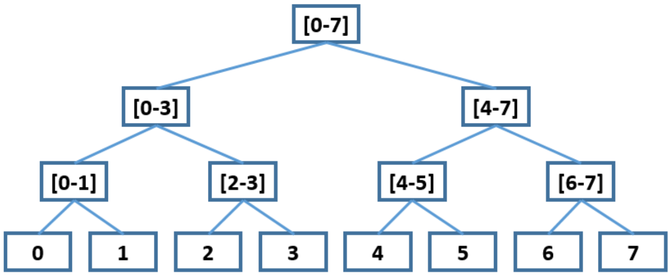

Figure 1: An example of DGH for x-coordinate.

C. Loss Metric

In this paper, we quantify the loss using similar metric proposed in [10].

Definition 2. The information loss incurred during the generalization and suppression while replacing node$_i$ with node$_j$ in DGH $H_{\mathcal{A}}$ is defined as

$$LS(\text{node}_i, \text{node}_j) = \log_2 LF(\text{node}_j) - \log_2 LF(\text{node}_i) \text{ bits}, \quad (1)$$

where $LF(.)$ function returns the number of leaves in the subtree generated by a node and $LS(.)$ function returns the loss incurred by generalization of the nodes.

Example 2. Consider the DGH given in Fig. 1 the loss incurred while generalizing node $[4-5]$ to $[4-7]$ can be calculated as $\log_2 4 - \log_2 2 = 2$ bits.

While generalizing two nodes, it is necessary to find the lowest common ancestor (LCA). The definition of LCA is given in Definition 3.

Definition 3. The LCA of nodes node$_i$ and node$_j$ in $H_{\mathcal{A}}$ is defined as the lowest common parent root of the two nodes. Function LCA returns the LCA.

The total loss incurred by generalizing node$_i$ and node$_j$ in $H_{\mathcal{A}}$ with their LCA node$_p$ can be calculated as

$$LS(\text{node}_i + \text{node}_j, \text{node}_p) = LS(\text{node}_i, \text{node}_p) + LS(\text{node}_j, \text{node}_p). \tag{2}$$

This value for generalization of trajectory tr to achieve the anonymized trajectory $\overline{tr}$ with respect to attribute $\mathcal{A}$ can be calculated as

$$LS(\overline{tr}, \mathcal{A}) = \sum_{i=1}^{|\overline{tr}|} LS(tr_i.\mathcal{A}, \overline{tr}_i.\mathcal{A}). \tag{3}$$

where $tr_i.\mathcal{A}$ indicates the i-th location of the trajectory tr with respect to the attribute $\mathcal{A}$. Here, A could denote x-coordinate, y-coordinate, or time. Similarly, the total loss with respect to an attribute $\mathcal{A}$ in an anonymized dataset $\overline{T}$ can be computed as

$$LS(\overline{T}, \mathcal{A}) = \sum_{\overline{tr} \in \overline{T}}^{|\overline{T}|} LS(\overline{tr}, \mathcal{A}) \tag{4}$$

The problem we seek to answer in this paper is formally presented in Problem 1 as follows.

Problem 1. *Given a trajectory dataset T, a privacy requirement k, quasi identifiers x-coordinate, y-coordinate, and time, how to generate an anonymized dataset $\overline{T}$ which achieves the k-anonymity privacy metric and minimizes the total loss with respect to all the quasi-identifiers formulated as*

$$Minimize\{LS(\overline{T}, x) + LS(\overline{T}, y) + LS(\overline{T}, t)\}. \tag{5}$$

III. Proposed Approach

Our proposed anonymization framework consists of a robust alignment technique and a machine learning approach for clustering the trajectory datasets which are presented in this section.

A. Alignment

The process of alignment is defined as finding the best match between two trajectories in order to minimize the overall cost of generalization and suppression. The process of alignment between two trajectories has been studied in different domains mostly referred to as sequence alignment (SA). In this paper, we incorporate a multiple SA technique called progressive SA [11].

1) Progressive Sequence Alignment: The progressive SA is commonly used for SA of a set of protein sequences. Progressive SA is a heuristic approach for multiple SA. As part of the algorithm, pairwise alignment of the trajectories is also required. We use dynamic SA for pairwise alignment of trajectories in progressive SA. Dynamic SA is based on dynamic programming and commonly used in DNA SA [12], [13]. Fig. 2 explains an example of how the progressive SA works for four hypothetical sequences $tr_a = \{a_1, a_2, a_3, a_4\}$, $tr_b = \{b_1, b_2\}$, $tr_c = \{c_1, c_2, c_3\}$ and $tr_d = \{d_1, d_2\}$ to generate the resulting aligned trajectory $tr_r = \{r_1, r_2, r_3, r_4\}$. The longest path ($tr_a$) is chosen as the basis and it is aligned with a randomly chosen trajectory (tr_b). The pairwise alignment process is implemented using dynamic SA. Then, the resulting trajectory is aligned with a third trajectory. The process continues until all the trajectories are aligned. Instead of choosing the trajectories randomly during the progressive SA, the algorithm can choose the trajectory resulting in the lowest loss during the alignment. In Fig. 2, the way trajectory elements are located with respect to the longest path is referred to as the structure of the shorter path and also the spaces indicate the suppression operation during the alignment.

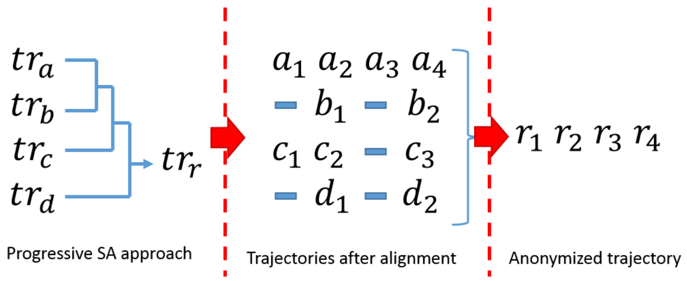

Figure 2: An example of the progressive SA.

The dynamic SA algorithm is formally represented in Algorithm 1. Dynamic SA is based on dividing the problem of finding the best SA to subproblems and storing the solutions of subproblems in a table or matrix referred to as $SAmatrix$ in the pseudocode. The objective is to achieve the minimal cost for SA. As before, the cost of alignment refers to the loss incurred during the alignment for different attributes of the sequence which are x-coordinate, y-coordinate, and the time of the query. A subproblem generation for matching the first to j-th element of tr_1 ($tr_1 = \{p_1, , p_2, ..., p_j\}$) with the first to i-th element of tr_2 ($tr_2 = \{q_1, , q_2, ..., q_i\}$) can be given as 1) match p_j and q_i; find the optimal alignment for $tr_1 = \{p_1, , p_2, ..., p_{j-1}\}$ and $tr_2 = \{q_1, , q_2, ..., q_{i-1}\}$ 2) suppress p_j; find the optimal alignment for $tr_1 = \{p_1, , p_2, ..., p_{j-1}\}$ and $tr_2 = \{q_1, , q_2, ..., q_i\}$ 3) suppress q_i; find the optimal alignment for $tr_1 = \{p_1, , p_2, ..., p_j\}$ and $tr_2 = \{q_1, , q_2, ..., q_{i-1}\}$.

The algorithm starts by creating a $(m + 1) \times (n + 1)$ matrix ($SAmatrix$), where m and n denote the length of the trajectories. The matrix will be used to store the minimum cost of each cell of the grid. Moreover, a list called *code* stores how the cells have been reached. Cell $[j+1, i+1]$ can be reached from three cells $[j, i+1]$, $[j+1, i]$, $[j, i]$. Each path corresponds to one of the subproblems explained. After finding all the values of the matrix and tracing back the list *code*, the outputs of the algorithm would be the value of cell $[m, n]$ which is the minimum value of the total loss ($TotLoss$) required for the dynamic SA, the aligned trajectory ($GenTraj$), and the structure of the shorter path compared to the longer path as $ShoTrajStr$.

Algorithm 1: DynamicSA(tr_1, tr_2, H_x, H_y, H_t).

```
Required variables: $tr_1 = \{p_1, , p_2, ..., p_m\}$,
  $tr_2 = \{q_1, , q_2, ..., q_n\}$, $H_x$, $H_y$, $H_t$
$SAmatrix \leftarrow$ np.zeros($[m+1, n+1]$)
for $i$ in range($m$) do
    $Loss \leftarrow LS(p_i.x,\ rt(H_x)) + LS(p_i.y,\ rt(H_x))$
        $+LS(p_i.t,\ rt(H_t))$
    $SAmatrix[i+1, 0] \leftarrow SAmatrix[i, 0] + Loss$
end
for $i$ in range($n$) do
    $Loss \leftarrow LS(q_i.x,\ rt(H_x)) + LS(q_i.y,\ rt(H_x))$
        $+LS(q_i.t,\ rt(H_t))$
    $SAmatrix[0, i+1] \leftarrow SAmatrix[0, i] + Loss$
end
$options \leftarrow$ np.zeros(3)
$code \leftarrow$ list()
for $i$ in range($m$) do
    for $j$ in range($n$) do
        $Loss \leftarrow$ loss incurred by generalizing $p_i$ and $q_j$
        $options[0] \leftarrow SAmatrix[i, j] + Loss$
        $Loss \leftarrow$ loss incurred by suppressing $q_j$
        $options[1] \leftarrow SAmatrix[i+1, j] + Loss$
        $Loss \leftarrow$ loss incurred by suppressing $p_i$
        $options[2] \leftarrow SAmatrix[i, j+1] + Loss$
        $BestOption \leftarrow$ np.argmin($options$)
        $code$.append(index of option with minimum
          value)
    end
end
$TotLoss \leftarrow SAmatrix[m, n]$
$GenTraj \leftarrow$ trace back the $code$ to generate the
  aligned trajectory
$ShoTrajStr \leftarrow$ trace back the $code$ to find out
  structure of shorter trajectory while alignment
Return $GenTraj, ShoTrajStr, TotLoss$
```

B. Clustering

Clustering can be seen as a search for hidden patterns that may exist in datasets. In simple words, it refers to grouping data entries in disjointed clusters so that the members of each cluster are very similar to each other.

1) k'-means Clustering Approach: k'-means algorithm [14] is an attractive clustering algorithm currently used in many applications particularly in data analysis and pattern recognition [15], [16]. The main advantage of k'-means algorithm is its simplicity and fast execution time. The reason behind using a prime notation on top of the variable k is to avoid any confusion between the meaning of "k" in the clustering algorithm and the k used in the definition of k-anonymity addressed before.

The algorithm aims to partition the input dataset into k' clusters. The only inputs to the algorithm are the number of clusters k' and the dataset. Clusters are represented by adaptively-changing cluster centres. The initial values of the cluster centres are chosen randomly. In each stage, the algorithm computes the squared distance of data from the centroids and partition them based on the nearest centroid to each data. The algorithm continues the same process until the values of centroids no longer change. The k'-means algorithm is guaranteed to converge [17].

In the rest of this section, we explain how the k'-means algorithm can be applied to trajectory datasets to increase the privacy of users while publishing the data.

Lemma 1. *The total loss incurred by generalizing $node_i$ and $node_j$ with respect to $H_{\mathcal{A}}$ can be calculated as*

$$LS(node_i,node_j) = \\ |LS(node_i, RT(H_{\mathcal{A}})) - LS(node_j, RT(H_{\mathcal{A}}))|. \tag{7}$$

Lemma 1 indicates that the loss incurred by generalizing two nodes is equal to the difference of losses incurred by their suppression. As before, for any clustering outcome of data, assume that $cluster$ is a two-dimensional list in which j-th element of the list returns the IDs of the trajectories in the j-th cluster. Moreover, we denote the j-th cluster head after generalization and suppression for all the trajectories by h_j. Therefore, the total loss can be written as

$$\begin{aligned}\text{Total loss} &= LS(\overline{T}, x) + LS(\overline{T}, y) + LS(\overline{T}, t) \\ &= \sum_{j=0}^{k-1} \sum_{tr \in cluster[j]} (LS(h_j.x, tr.x) \\ &\qquad + LS(h_j.y, tr.y) + LS(h_j.t, tr.t)).\end{aligned} \tag{8}$$

As explained in the equation (5), the objective of clustering algorithms is to minimize this equation. Therefore, using Lemma 1 the equation (8) can be written as

$$\text{Total loss} = \tag{9}$$

$$\begin{aligned}&\sum_{j=0}^{k-1} \sum_{tr \in cluster[j]} (|LS(h_j.x, RT(H_x)) - LS(tr.x, RT(H_x)| \\ &\quad + |LS(h_j.y, RT(H_y)) - LS(tr.y, RT(H_y)| \\ &\quad + |LS(h_j.t, RT(H_t)) - LS(tr.t, RT(H_t))|.\end{aligned} \tag{10}$$

Rearranging equation (9), the objective equation can be found by minimizing equation (6). This can be done by maximizing part B and minimizing part A. Since the cluster heads are generated based on the clustering algorithm, they cannot be used as part of the optimization process. Therefore, we aim at minimizing part A of the equation (6).

Part A in the equation (6) refers to finding the total distance of each trajectory from DGH root of the attributes. Therefore,

$$\text{Total loss} = \underbrace{\sum_{i=1}^{|T|}(LS(tr_i.x, RT(H_x)) + LS(tr_i.y, RT(H_y)) + LS(tr_i.t, RT(H_t)))}_{\text{A}} - \underbrace{\left(\sum_{i=1}^{|cluster|}\sum_{j=1}^{|cluster[i]|}(LS(h_j.x, RT(H_x)) + LS(h_j.y, RT(H_y)) + LS(h_j.t, RT(H_t)))\right)}_{\text{B}}. \tag{6}$$

for each trajectory a three dimensional vector $< d_x, \, d_y. \, d_t >$ is constructed where d_x, d_y, d_t store the loss incurred by generalizing the x-coordinate, y-coordinate, and time, respectively. Having distance of all the points from the roots, we cluster the trajectories using the k'-means algorithm. The algorithm clusters trajectories with a similar loss from the root in the same group. This process is particularly important as trajectory datasets usually include trajectories as short as one query to trajectories with hundreds of queries.

k'-means algorithm clusters the trajectories without any constraint on the minimum number of trajectories that needs to be in each cluster. For more sensitive applications, a heuristic approach can be applied on top of the k'-means to make sure that the all the clusters include at least k' trajectories.

IV. Experiments

In our experiment, we use the data collected by Geolife project [18]. The geolife dataset includes the GPS trajectories of 182 users from April 2007 to August 2012 in Beijing, China. The dataset entails $17,621$ trajectories with a total distance of $1,292,951km$. Each entry of data is represented with coordinates and the time stamp in which the query has happened. We have conducted our experiments on $1km \times 1km$ central part of the Beijing map with the resolution of $0.01km \times 0.01km$ for each grid cell. The location privacy requirement (k) of the users are investigated for the values 2, 5, 10 and 15. The experiments are performed on a PC with a 3.40GHz core-i7 Intel processor, 64-bit Windows 7 operating system, and 8.00GB of RAM. Python program is used to implement the algorithms.

It must be noted that the dataset includes trajectories as large as hundreds of queries and as small as a single query from the location-based service provider. Therefore, matching such variant length trajectories would impose a large loss even for the best possible match of the sequences. Incurred loss of the proposed framework is demonstrated in Fig. 3. The x-axis indicates privacy requirement for the dataset (k) and the y-axis indicates the total loss incurred which includes the loss while applying generalization and suppression on x-coordinate, y-coordinate, and the time of the query. Furthermore, the maximum possible incurred loss which refers to suppressing all the trajectories is shown by a green dashed line on the graphs. As expected, increasing the value of k results in a higher incurred loss due to having larger cluster sets which mean alignment of a higher number of trajectories in each cluster.

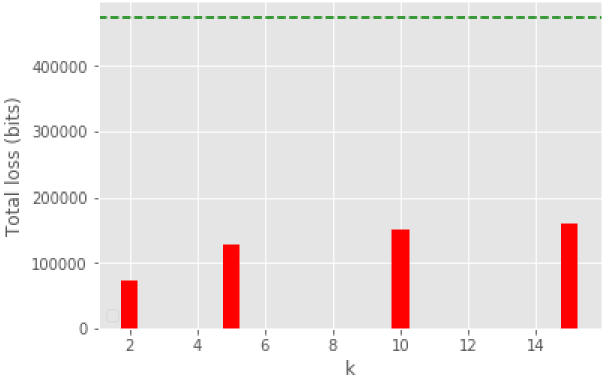

Figure 3: Incurred loss of the proposed algorithm based on k'-means clustering.

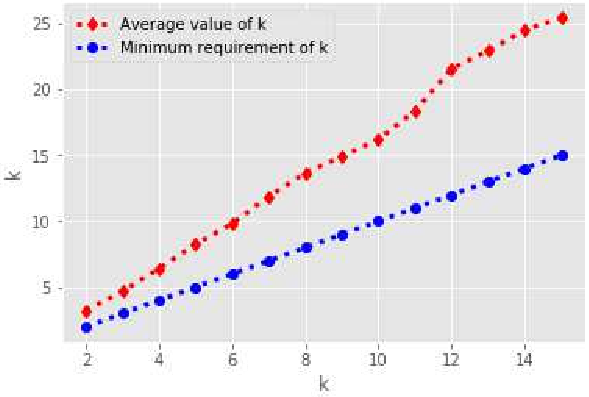

Figure 4: Average value of k achieved applying k'-means algorithm.

Due to lack of any constraint on k'-means algorithm, some of the users may experience privacy metric lower than k-anonymity as it is the case in mobile networks in which some of the users may experience a lower quality of service. Fig. 4 indicates the average value of k achieved while applying the k'-means algorithm. It can be seen in the figure that on average individuals are achieving a privacy level higher than k applying the proposed approach. The value of the average

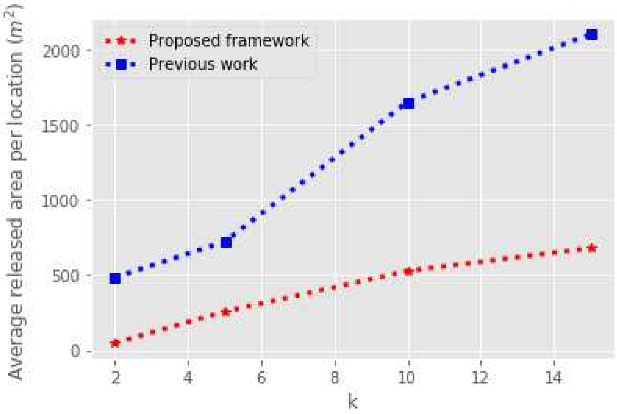

Figure 5: Comparison of our proposed framework with the previous work in [8].

gets even better as the value of k increases.

Fig. 5 indicates the result of comparison between our proposed anonymization technique and the recent generalization method proposed in [8]. The authors in [8], attempt to minimize the incurred loss during the anonymization by sorting out the spatiotemporal locations in the time domain and applying a heuristic approach. Note that the aim of any anonymization approach is to maximize utility while preserving the privacy of users. Utility in generalization techniques refers to the area released for locations in the dataset. Therefore, to have a fair comparison, we compare our work with the approach proposed in [8] based on the average released area for locations. It can be seen from the figure that our proposed algorithm can significantly increase the utility of the generalization approach. In other words, the anonymized dataset has on average smaller released area per location while preserving the privacy of users.

V. Conclusion

In this paper, we have developed a method to preserve the privacy of users while publishing the spatiotemporal trajectories. The proposed approach incorporates multiple sequence alignment for anonymization in addition to developing a technique that enables the use of machine learning methods for clustering in the context of privacy. We implemented the clustering based on k'-means clustering algorithm and applied it on to Geolife dataset.

References

[1] A. Government, "New australian government data sharing and release legislation," 2018.

[2] B. Liu, W. Zhou, T. Zhu, L. Gao, and Y. Xiang, "Location privacy and its applications: a systematic study," *IEEE access*, vol. 6, pp. 17 606–17 624, 2018.

[3] S. Shaham, M. Ding, B. Liu, Z. Lin, and J. Li, "Privacy preservation in location-based services: A novel metric and attack model," *arXiv preprint arXiv:1805.06104*, 2018.

[4] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[5] A. Tamersoy, G. Loukides, M. E. Nergiz, Y. Saygin, and B. Malin, "Anonymization of longitudinal electronic medical records," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 3, pp. 413–423, 2012.

[6] 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 *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1241–1250.

[7] Y. Dong and D. Pi, "Novel privacy-preserving algorithm based on frequent path for trajectory data publishing," *Knowledge-Based Systems*, vol. 148, pp. 55–65, 2018.

[8] M. Gramaglia, M. Fiore, A. Tarable, and A. Banchs, "Towards privacy-preserving publishing of spatiotemporal trajectory data," *arXiv preprint arXiv:1701.02243*, 2017.

[9] M. Terrovitis, G. Poulis, N. Mamoulis, and S. Skiadopoulos, "Local suppression and splitting techniques for privacy preserving publication of trajectories," *IEEE Trans. Knowl. Data Eng*, vol. 29, no. 7, pp. 1466–1479, 2017.

[10] V. S. Iyengar, "Transforming data to satisfy privacy constraints," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 279–288.

[11] B. Chowdhury and G. Garai, "A review on multiple sequence alignment from the perspective of genetic algorithm," *Genomics*, 2017.

[12] X. Chen, C. Wang, S. Tang, C. Yu, and Q. Zou, "Cmsa: a heterogeneous cpu/gpu computing system for multiple similar rna/dna sequence alignment," *BMC bioinformatics*, vol. 18, no. 1, p. 315, 2017.

[13] Q. Le, F. Sievers, and D. G. Higgins, "Protein multiple sequence alignment benchmarking through secondary structure prediction," *Bioinformatics*, vol. 33, no. 9, pp. 1331–1337, 2017.

[14] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[15] S. K. Pal and P. P. Wang, *Genetic algorithms for pattern recognition*. CRC press, 2017.

[16] S. Shaham, M. Kokshoorn, Z. Lin, M. Ding, and Y. Wu, "Raf: Robust adaptive multi-feedback channel estimation for millimeter wave mimo systems," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.

[17] A. Fischer and D. Picard, "Convergence rates for smooth k-means change-point detection," *arXiv preprint arXiv:1802.07617*, 2018.

[18] Y. Zheng, X. Xie, and W.-Y. Ma, "Geolife: A collaborative social networking service among user, location and trajectory." *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, 2010.