International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Location Based Privacy Preserving Using Machine Learning Approach

Roshini Singh[1], Guguloth Kiranmai[2], A. Sangeetha[3]

[1,2]Student, Chaitanya Bharathi Institute of Technology, Hyderabad
[3]Assistant Professor, CSE, Chaitanya Bharathi Institute of Technology, Hyderabad

Abstract
Location-based privacy preservation is crucial in the digital age due to the widespread use of location-based services and growing concerns about individual privacy. Despite the use of k-anonymity measures, current systems face challenges, particularly the risk of re-identification, especially when attackers have additional contextual information. These systems also suffer from information loss, leading to a significant decrease in data utility. Striking the right balance between privacy and data utility remains a prominent challenge in the field of location-based privacy preservation. To address the existing gaps and challenges in privacy-preserving location data publishing, a robust framework termed Location anonymization framework is being introduced which consists of three methods Generalization, Sequence alignment and Clustering. DGH Trees are being used to minimize the information loss and also maintains a balance between privacy and utility. To enhance privacy preservation in overly sensitive datasets, a modified version of the k-means algorithm is being put forth. Moreover, to improve the alignment process, the more efficient iterative multisequence alignment is being opted for over the progressive counterpart within this framework. The aim is to achieve a more balanced trade-off between privacy and utility in location data publishing.

Keywords: Domain Generalization Hierarchy, Location Based Privacy Preserving, Alignment, Clustering, Location Anonymization Framework, Data Privacy, Quasi Identifiers, Information Loss, Data Security, Privacy Trade-off metric, Pairwise Alignment of Residue Patterns.

1. Introduction

In the era of digitalization, location-based services have become an integral part of our daily lives, providing us with a wide range of applications such as navigation, ride-sharing, and social networking. However, the widespread use of these services has raised significant concerns about individual privacy, particularly in the context of location-based data. Despite the use of k-anonymity measures, current systems face challenges, particularly the risk of re-identification, especially when attackers have additional contextual information. These systems also suffer from information loss, leading to a significant decrease in data utility.

To address the existing gaps and challenges in privacy-preserving location data publishing, this report introduces a robust framework that consists of three methods: Generalization, Sequence alignment, and Clustering. The framework employs Domain Generalization Hierarchy (DGH) Trees for Generalization to balance privacy and utility. To enhance privacy preservation in overly sensitive datasets, a modified

version of the k-means algorithm is used. Additionally, the framework utilizes iterative multisequence alignment to improve the alignment process, which is more efficient than the progressive counterpart.
The aim of this framework is to achieve a more balanced trade-off between privacy and utility in location data publishing. By leveraging this comprehensive anonymization framework, the report aims to mitigate potential privacy breaches stemming from the exploitation of location-based data while preserving the utility and integrity of the data for subsequent analysis or applications.

1.1. The main features of this research work are listed as follows

- **Enhanced Privacy Preservation:** The primary objective is to improve privacy preservation in location data publishing. By introducing a modified version of the k-means algorithm, the framework aims to anonymize location data effectively while maintaining the statistical properties of the dataset.
- **Efficient Alignment:** By selecting the iterative multisequence alignment over the progressive counterpart, the methodology aims to enhance the efficiency of the alignment process. This choice is intended to improve the scalability and computational efficiency of the framework while ensuring accurate alignment of location sequences.
- **Balanced Trade-off:** The overarching goal is to achieve a balanced trade-off between privacy and utility in location data publishing. This involves finding a middle ground where the anonymized data sufficiently protects individual privacy while still providing valuable insights and utility for analysis and decision-making purposes.
- **Robust Framework:** The methodology aims to establish a robust framework for location anonymization. This includes addressing existing gaps and challenges in privacy-preserving location data publishing by incorporating advanced techniques and algorithms to ensure the effectiveness and reliability of the framework.

2. Related works

Bo Liu et.al.[1] presented a novel privacy-preserving framework that enables the secure release of sensitive spatiotemporal trajectory datasets containing individuals' movements and locations over time. This framework leverages Machine Learning-based Anonymization (MLA) techniques, integrating advanced clustering and alignment methodologies to balance user privacy and data utility. It employs a modified k-means clustering algorithm tailored for spatiotemporal data to achieve k-anonymity and a sequence alignment technique to minimize information loss during anonymization. Experiments on real-life GPS trajectory datasets demonstrated the framework's superior performance in preserving spatial utility while safeguarding user privacy.
Anup Maurya1 et.al.[2] presented a comprehensive evaluation and comparison of three prominent anonymization algorithms - Top-Down, Mondrian, and an Improved Mondrian algorithm - focusing on information loss and execution time. It aims to identify the most effective approach for balancing robust privacy protection and data utility. The Improved Mondrian algorithm outperformed the others, demonstrating superior privacy protection capabilities with minimal information loss, preserving analytical value, and exhibiting faster execution times for efficient anonymization of large datasets. The findings contribute to the field of privacy-preserving data publishing by providing insights into the strengths and limitations of these anonymization approaches.
Dhananjay M. Kanade1 et.al.[3] evaluated and compared three prominent anonymization algorithms: Top-Down, Mondrian, and an Improved Mondrian algorithm, focusing on information loss and execution time. Information loss directly impacts data utility, while execution time reflects computational efficiency,

crucial for large datasets. The Improved Mondrian algorithm outperforms the others, demonstrating superior privacy protection capabilities by effectively anonymizing data while minimizing information loss, preserving analytical value. Additionally, it exhibits faster execution times, enabling efficient anonymization of large datasets. The study quantifies information loss and execution time, aiming to identify the most effective approach balancing privacy protection and data utility. The findings contribute to privacy-preserving data publishing by providing insights into the strengths and limitations of these anonymization approaches, guiding researchers, analysts, and organizations in implementing effective strategies.

Zhenpeng Liu et.al.[4] proposed a methodology which offers a comprehensive, multi-layered approach to safeguarding user location privacy in continuous location services. It integrates variable-order Markov modeling to capture probabilistic dependencies in location data, enabling granular understanding of user movements. Differential privacy techniques introduce controlled noise, obfuscating individual-level information while preserving data utility. Cache-based mechanisms securely manage location data, minimizing direct access risks. K-anonymity techniques anonymize location information by grouping individuals into larger clusters, thwarting identification of specific users. By synergistically combining data modeling, noise injection, secure data management, and anonymization, this holistic approach enhances user privacy while maintaining location data's accuracy and utility. This methodology enables continued development of valuable location-based services while respecting users' privacy rights and concerns.

Long LI1,2 et.al.[5] presented a novel methodology for anonymizing spatiotemporal trajectory datasets, addressing user privacy while preserving data utility. This approach leverages machine learning techniques, integrating clustering models to group similar trajectories, dynamic sequence alignment to identify and preserve significant patterns while obscuring individual details, and data generalization to transform precise location information into generalized representations. Through the synergistic integration of these techniques, the methodology offers a robust anonymization framework tailored for spatiotemporal data. By leveraging clustering, alignment, and generalization, it addresses privacy preservation while maintaining the analytical value of trajectory datasets. This novel approach aims to strike a balance between user privacy and data utility, enabling responsible utilization of location data while upholding individuals' privacy rights.

2.1.Research Objectives

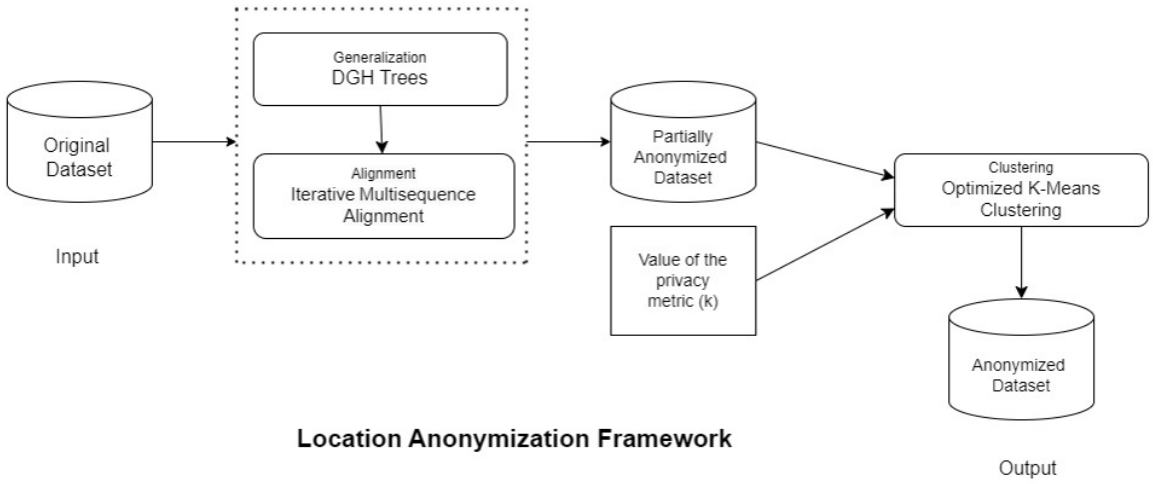
Based on the detailed description of your proposed anonymization framework, the following could be potential research objectives for your project:

1. Develop an effective and efficient generalization technique using DGH trees to anonymize location-based datasets while preserving geographic patterns and context.
2. Investigate and implement advanced iterative multisequence alignment algorithms, such as the Pairwise Alignment of Residue Patterns (PRRP), to refine and enhance the alignment of anonymized sequences, minimizing information loss.
3. Explore and optimize the K-Means clustering algorithm for the anonymized dataset, incorporating techniques such as density-based initial centroid selection and principal component analysis, to achieve optimal partitioning of the data while preserving inherent structures and patterns.
4. Evaluate the trade-off between the degree of anonymization achieved by the proposed framework and the utility of the resulting dataset for various applications, such as location-based services, urban planning, or transportation analysis.

5. Conduct comprehensive experiments and comparative analyses to assess the performance of the proposed anonymization framework in terms of privacy protection, data utility preservation, and computational efficiency, benchmarking against existing anonymization techniques.
6. Develop robust metrics and evaluation methodologies to quantify the level of privacy protection achieved by the anonymization framework, considering various privacy models and attack scenarios.
7. Investigate the potential impact of the anonymized datasets on downstream machine learning models or data analytics tasks, evaluating the trade-offs between privacy protection and model performance.

These objectives could guide the research efforts, experimentation, and evaluation of your proposed anonymization framework, contributing to the advancement of privacy-preserving data processing techniques for location-based datasets.

3. Proposed Methodology



Location Anonymization Framework

The proposed anonymization framework is a comprehensive and multi-staged process that aims to safeguard individual privacy while preserving the utility of location-based datasets containing quasi-identifiers, which are attributes that, when combined, could potentially lead to the re-identification of individuals. The framework commences with the generalization of the input dataset through the employment of DGH trees, hierarchical data structures that provide a systematic and organized representation of geographic entities at varying levels of granularity, whereby specific, granular location data points are strategically substituted with their corresponding generalized counterparts, obfuscating potentially identifying information while preserving the overall geographic context and patterns. The generalized output is then utilized as input for the iterative multisequence alignment stage, which employs sophisticated algorithms such as the Pairwise Alignment of Residue Patterns (PRRP) to refine and enhance the alignment of the anonymized sequences, iteratively identifying and realigning regions exhibiting high similarity to minimize information loss and preserve inherent patterns and homologies. The partially anonymized dataset subsequently serves as input for the final stage of the framework: optimized K-Means clustering, which employs an enhanced variant of the K-Means algorithm, strategically selecting initial cluster centroids using techniques such as density-based methods or principal component analysis, and iteratively refining these centroids and reassigning data points to their nearest cluster centroid, minimizing intra-cluster distances and maximizing inter-cluster distances until

convergence, yielding an optimal partitioning of the data into distinct clusters. The culmination of this multi-staged architecture is the generation of a fully anonymized dataset, wherein individual privacy is safeguarded through the judicious application of generalization, sequence alignment, and clustering techniques, while retaining the inherent patterns and structures within the data, thereby preserving its utility.

3.1.Modules

Python

Python is a popular programming language used for machine learning, especially due to its easy to use nature, adaptability and extensive range of libraries and frameworks designed to support machine learning. Python is the recommended language because it offers a strong framework for creating, training, and deploying complicated neural networks thanks to packages like TensorFlow and PyTorch.

Folium

Folium is a Python library for visualizing geospatial data using interactive Leaflet maps. It provides an easy-to-use interface for creating, customizing, and embedding maps in Python applications. Folium is valuable for data scientists, geospatial analysts, and developers working with spatial data, offering an intuitive API and seamless integration with libraries like pandas and geopandas.

OS

The Python os module offers an OS-agnostic interface for interacting with the operating system. It provides functions for process management, file/directory operations, and environment variable handling. It enables file permissions, modifications, attribute checks, directory creation, environment variable manipulation, process control, and system command execution. The os module provides a portable way to manage the file system, administer systems, and control processes from Python.

Anytree:

Anytree is a Python library offering a flexible implementation of tree data structures. It enables easy creation, modification, and traversal of various tree types like general and binary trees. Anytree is invaluable for working with hierarchical data across domains, providing an intuitive API and robust features for tree-related operations.

Matplotlib

Matplotlib is a versatile Python visualization library providing powerful charting capabilities. Its user-friendly interface, seamless integration with NumPy/Pandas, and extensive customization options enable efficient data exploration and visualization across domains. Matplotlib supports multiple output formats, making it ideal for publications and presentations, establishing itself as an essential data visualization tool in Python.

NumPy

NumPy provides support for multi-dimensional arrays, matrices, and mathematical functions for efficient array manipulation in Python. Its key features include ndarrays, broadcasting, and array operations. Optimized in C, NumPy is essential for scientific computing, data analysis, and machine learning, complementing libraries like Pandas and Matplotlib.

3.2.Work Flow

INPUT: Original Dataset.

OUTPUT: Anonymized Dataset.

METHOD:

- Collection of necessary location-based datasets containing Taxi ID, Longitude, Latitude, Date, and Time columns.
- First stage is Generalization, Dividing the geographic area into grid cells and representing each cell with a generalized value.
- Creation of a tree structure where each generalized grid cell value is represented as a parent node, and the data points (latitude values) within that cell are represented as child nodes.
- Performing Iterative Multiple Sequence Alignment (MSA) on the parent node values in the generalised tree.
- Applying modified version of k-means clustering on the aligned sequences where we get the final output.

3.2.1 Generalization (DGH Trees)

DGH Tree is a Domain Generalization Hierarchy Tree which is used to provide generalisation to the dataset which should be in hierarchal structure.

To Implement DGH Tree Generalization we follow the steps:

3.2.1.1.Visualization of the dataset

Folium is a Python library that allows users to create interactive maps from data manipulated in Python.we have used folium module for the visualization of dataset with latitude and longitude.

3.2.1.2.Formation of Grid

Next step in the generalization is to form the grid. We have formed 4x8 (Latitude x Longitude) grid with latitude as rows and longitude as columns by importing NumPy and pyplot modules.

3.2.1.3.Forming a tree structure with the Grid

Next step in the generalization is to form the tree structure. We have used anytree module for forming a tree structure.

3.2.1.4.Anonymization of tree

Tree is anonymized in such a way that latitude and longitude values are represented with the generalized values.

3.2.2. Alignment (Iterative Multi sequence Alignment)

When it comes to information loss during multiple sequence alignment, iterative refinement methods are generally considered better than progressive alignment methods.

Progressive methods, while computationally efficient, can accumulate errors during the alignment process, potentially leading to information loss.

Iterative refinement methods aim to reduce this information loss by refining the initial alignment iteratively. One popular iterative refinement method is PRRP (Pairwise Alignment of Residue Patterns).

3.2.2.1.Working of PRRP (Pairwise Alignment of Residue Patterns).

1. Perform an initial progressive alignment on the input sequences.
2. Identify residue patterns (regions with high similarity) in the initial alignment.
3. For each identified pattern:

a. Realign the regions corresponding to the pattern using a pairwise alignment method.

b. If the new alignment for the pattern region is better than the original, update the initial alignment with the new alignment.

4. Repeat steps 2 and 3 until there is no further improvement in the alignment score.

3.2.3. Clustering (Modified version of K-means)

Clustering algorithms aim to partition the data into distinct groups or clusters, where the data points within a cluster are more similar to each other than to those in other clusters.

We have implemented two clustering techniques

1. Heuristic clustering
2. Optimized K-Means clustering

3.2.3.1. Heuristic Clustering:

The heuristic algorithm is often applied for optimizing different objective functions, however, with a similar structure. We have used this approach as benchmark to compare our proposed algorithm.

The intuition behind the heuristic algorithm is to form the clusters by sequentially adding the most suitable trajectory that minimizes the total loss incurred by generalization and suppression.

3.2.3.2. Optimized K-means Clustering Algorithm:

The K-Means algorithm initializes centroids randomly and iteratively assigns data points to the closest centroid, updating the centroids based on the mean of the assigned points. In this case, the algorithm converged to a solution where all data points are in one cluster. This approach first runs the standard K-Means algorithm and then iteratively refines the clustering by removing clusters with at least k trajectories before reassigning the remaining points to the closest centroids. In this scenario, the iterative process did not lead to a different clustering outcome, resulting in all points being in the same cluster for both methods.

3.3.Methodologies

3.3.1. Generalization (DGH Trees)

3.3.1.1. Data_visualization Function

This function provides a simple way to visualize geographic data from a CSV file on an interactive map, which can be useful for various applications, such as plotting locations of businesses, tracking delivery routes, or analyzing spatial patterns in data.

INPUT: Location based Dataset

OUTPUT: Map representing location data points

1.Import the required libraries (Pandas and Folium).

2. Read the CSV file into a Pandas DataFrame.

3. Get the latitude and longitude of the first point in the DataFrame.

4. Create a Folium map object centered at the first point's coordinates with an initial zoom level of 15.

5. Iterate through each row in the DataFrame.

6. For each row:

a. Get the latitude and longitude from the current row.

b. Create a map marker with the latitude and longitude.

c. Add the marker to the map.

7.Save the map as an HTML file named "map.html".

OUTPUT:

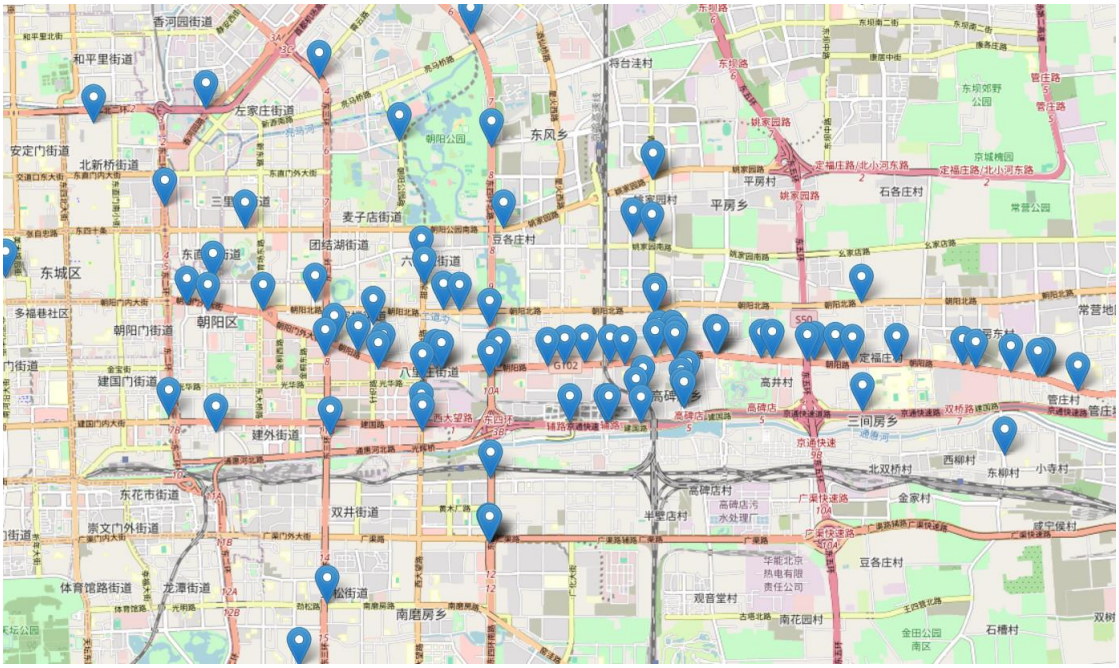


Fig 3.1: Visualization of T-Drive dataset

3.3.1.2.Grid_generation Function

This function allows you to visualize the distribution of data points within a geographical area by dividing it into a grid. The grid lines can help identify patterns or clusters in the data, making it useful for spatial analysis and data exploration.

INPUT: Location based Dataset

OUTPUT: Grid structure

1. Import required libraries: Pandas, NumPy, and Matplotlib.
2. Read the CSV file into a Pandas DataFrame.
3. Define the number of rows and columns for the grid.
4. Calculate the minimum and maximum values of longitude and latitude from the data.
5. Determine the step sizes for longitude and latitude based on the grid dimensions.
6. Create a list of dictionaries representing each grid cell with its row, column, and longitude/latitude ranges.
7. Convert the list of dictionaries into a Pandas DataFrame.
8. Create a figure and scatter plot for the data points.
9. Iterate through each row in the grid DataFrame:

a. Plot the grid cell boundaries as red dashed lines on the scatter plot.

10. Add labels, title, legend, and grid lines to the plot.
11. Display the plot.

OUTPUT:

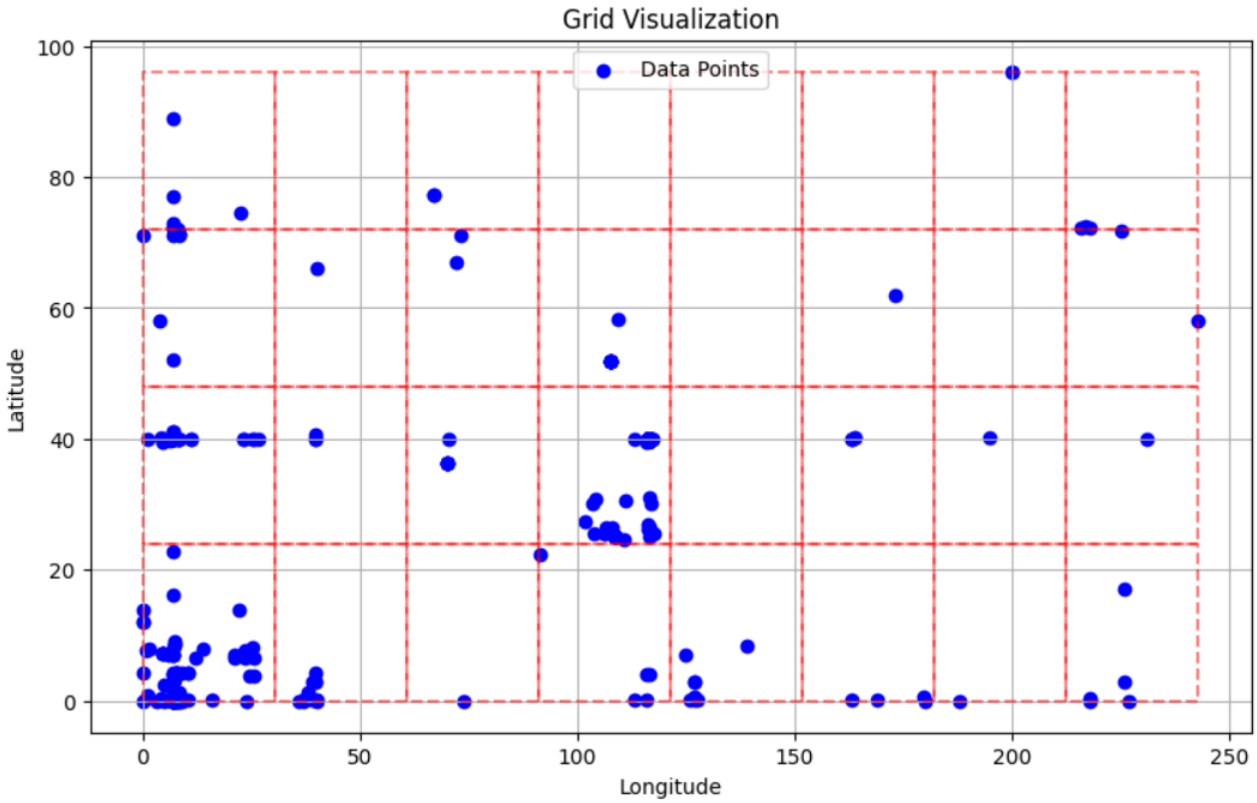


Fig 3.2: Grid Formation

3.3.1.3.Tree_generation Function

This function organizes the latitude and longitude data into a hierarchical tree structure, making it easier to visualize and analyze the distribution of data points within different ranges. The resulting text files can be used for further processing or visualization, if needed.

INPUT: Location based Dataset

OUTPUT: Tree structure

1. Define the number of rows and columns for latitude and longitude ranges.
2. Calculate the minimum and maximum values of latitude and longitude from the data.
3. Determine the step sizes for latitude and longitude based on the number of rows and columns.
4. Create dictionaries to store leaf nodes for latitude and longitude.
5. Iterate through the DataFrame:

a. Calculate the root latitude and longitude for each data point based on the step sizes.

b. Populate the leaf node dictionaries with the corresponding latitude and longitude values.

6. Create root nodes for latitude and longitude trees.
7. Populate the latitude tree structure:

a.Create child nodes for each root latitude.

b. If only one data point exists for a root latitude, create a leaf node.

c.Otherwise, create an intermediate root node and leaf nodes as children.

8. Populate the longitude tree structure (similar to latitude).
9. Define file paths for saving the output.
10. Define a function to save the tree structure to a file using the anytree library.

OUTPUT:

Lattitude

```
Latitude Range: (0.0, 72.08014)
├── Root: 36.04007
|   ├── Leaf: 39.92123
|   ├── Leaf: 39.97043
|   ├── Leaf: 39.91296
|   ├── Leaf: 39.85721
|   ├── Leaf: 39.84292
|   ├── Leaf: 39.85573
|   ├── Leaf: 39.86658
|   ├── Leaf: 39.99651
|   ├── Leaf: 39.89127
|   ├── Leaf: 39.89111
|   ├── Leaf: 39.89111
|   ├── Leaf: 39.89959
|   ├── Leaf: 39.91654
|   ├── Leaf: 39.92152
|   ├── Leaf: 46.4096
|   ├── Leaf: 46.4096
|   ├── Leaf: 46.4096
|   ├── Leaf: 46.4096
|   └── Leaf: 48.42295
└── Root: 72.08014
    ├── Leaf: 68.09484
    ├── Leaf: 68.09484
    ├── Leaf: 68.09484
    ├── Leaf: 68.09484
    ├── Leaf: 68.09484
    ├── Leaf: 72.08014
    └── Leaf: 72.08014
```

Longitude

```
Longitude Range: (0.0, 195.01217)
├── Root: 121.88260625
|   ├── Leaf: 116.51172
|   ├── Leaf: 116.47284
|   ├── Leaf: 116.48347
|   ├── Leaf: 116.42589
|   ├── Leaf: 116.35883
|   ├── Leaf: 116.41487
|   ├── Leaf: 116.46671
|   ├── Leaf: 116.47272
|   ├── Leaf: 116.48331
|   ├── Leaf: 116.48335
|   ├── Leaf: 116.48335
|   ├── Leaf: 116.48364
|   ├── Leaf: 116.51238
|   ├── Leaf: 27.53383
|   ├── Leaf: 21.7335
|   ├── Leaf: 21.38483
|   ├── Leaf: 26.5
|   └── Leaf: 14.7505
├── Root: 195.01217
|   ├── Leaf: 194.01067
|   └── Leaf: 195.01217
└── Root: 170.63564875
    ├── Leaf: 163.76429
    └── Leaf: 172.86782
```

Fig 3.3: Tree Formation

3.3.1.4.Anonymized_tree_generation Function

This function anonymizes the latitude and longitude data by grouping them into generalized ranges and organizing the ranges into a hierarchical tree structure. The resulting text files can be used for further processing or visualization while preserving the anonymity of the original data points.

INPUT: Location based Dataset

OUTPUT: Anonymized tree structure

1. Import required libraries: Pandas and anytree.
2. Read the CSV file into a Pandas DataFrame.
3. Define functions to generalize latitude and longitude values into ranges based on a bin size.
4. Define the number of rows and columns for the grid.
5. Calculate the minimum and maximum values of latitude and longitude from the data.
6. Determine the step sizes for latitude and longitude based on the number of rows and columns.
7. Create dictionaries to store leaf nodes for generalized latitude and longitude ranges.
8. Iterate through the DataFrame:

a. Calculate the root latitude and longitude for each data point based on the step sizes.

b. Generalize the latitude and longitude values into ranges using the provided functions.

c.Populate the leaf node dictionaries with the generalized ranges.

9. Create root nodes for latitude and longitude trees.
10. Populate the latitude tree structure:

a. Create child nodes for each root latitude range.

b. If only one generalized range exists for a root latitude, create a leaf node.

c. Otherwise, create an intermediate root node and leaf nodes as children for the generalized ranges.

11. Populate the longitude tree structure (similar to latitude).

12. Define file paths for saving the output.
13. Define a function to save the tree structure to a file using the anytree library.

OUTPUT:

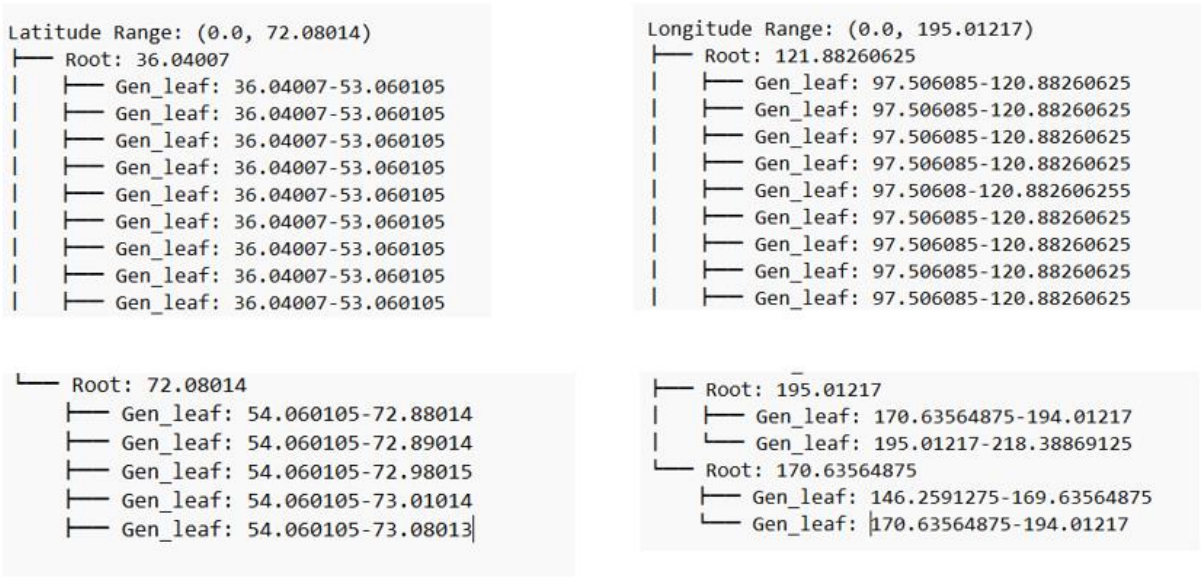


Fig 3.4: Anonymized Tree Structure

3.3.2. Iterative Multi-Sequence Alignment

3.3.2.1.PRRP Function

This Python code demonstrates how to represent latitude and longitude values as biological sequences using the Biopython library. The latitude and longitude values are first converted to fixed-length strings, and then these strings are treated as biological sequences. The sequences are aligned using Biopython's MultipleSeqAlignment module, which performs a multiple sequence alignment.

INPUT: List of latitude and Longitude values

OUTPUT: List of Sequence

1. DEFINE sequences as a list of latitude values
2. FIND the maximum length of the string representations of the latitude values
3. FOR each latitude value in sequences:

a. CONVERT the latitude value to a fixed-length string by padding with zeros if necessary

b. CREATE a SeqRecord object with the formatted string as the sequence, and appropriate ID and description

4. CREATE a MultipleSeqAlignment object with the list of SeqRecord objects
5. PRINT the alignment

OUTPUT:

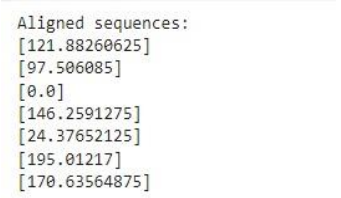


Fig 4: Sequence Alignment

3.3.3. Clustering

3.3.3.1.Heuristic_clustering Function

This function implements a heuristic approach to cluster trajectories in a dataset while trying to minimize information loss. The anonymized dataset includes the cluster assignments, which can be useful for further analysis or processing.

INPUT: Partially Anonymized Dataset

OUTPUT: Anonymized Dataset

1. Read the input dataset from a CSV file using Pandas.
2. Define a function align_trajectories that calculates the information loss when aligning two trajectories. (For simplicity, it returns a random value between 0 and 1.)
3. Define a function heuristic_clustering that takes the original dataset and the desired number of clusters k as input.

a. Initialize k empty clusters.

b. Randomly assign trajectories to the clusters.

c. Refine the clusters by iteratively moving trajectories to the cluster where they have the minimum total information loss when aligned with the existing trajectories in that cluster.

4. Define a function generate_anonymized_dataset that takes the clusters and the original dataset as input.

a. For each cluster:

i. For each trajectory in the cluster:

1. Calculate the information loss by aligning the trajectory with itself (assuming trajectories are anonymized in-place).
2. Add the cluster label to the trajectory.

Append the trajectory with the cluster label to the anonymized dataset.

b. Return the anonymized dataset and the total information loss.

5. Update the original dataset with the data from the CSV file.6. the heuristic_clustering function with the original dataset and the desired number of clusters k.
7. Convert the anonymized dataset to a Pandas DataFrame with an additional column for the cluster labels.
8. Save the anonymized dataset with the cluster labels to a new CSV file

OUTPUT:

Partially Anonymized dataset

	Taxi	Date	Time	Longitude	Latitude	Latitude_Root	Longitude_Root
0	1	2022008	56168	97.506085-120.88260625	36.04007-53.060105	36.04007	97.506085
1	1	2022008	56768	97.506085-120.88260625	36.04007-53.060105	36.04007	97.506085
2	1	2022008	56768	97.506085-120.88260625	36.04007-53.060105	36.04007	97.506085
3	1	2022008	57368	97.506085-120.88260625	36.04007-53.060105	36.04007	97.506085
4	1	2022008	57968	97.506085-120.88260625	36.04007-53.060105	36.04007	97.506085
..	...	...	...	...	...	...	...
583	1	8022008	54691	97.506085-120.88260625	36.04007-53.060105	36.04007	97.506085
584	1	8022008	55291	97.506085-120.88260625	36.04007-53.060105	36.04007	97.506085
585	1	8022008	55891	97.506085-120.88260625	36.04007-53.060105	36.04007	97.506085
586	1	8022008	56491	97.506085-120.88260625	36.04007-53.060105	36.04007	97.506085
587	1	8022008	57091	97.506085-120.88260625	36.04007-53.060105	36.04007	97.506085

Fig 3.5: Partially Anonymized Dataset

Anonymized dataset after Heuristic clustering

ID	Longitude Range	Latitude Range	Date	Time	Latitude	Longitude	Cluster
8130	97.506085-120.88260625	36.04007-53.060105	2008-02-04	18:38:47	36.04007	97.506085	0
3433	97.506085-120.88260625	36.04007-53.060105	2008-02-04	15:20:30	36.04007	97.506085	0
8131	97.506085-120.88260625	36.04007-53.060105	2008-02-07	01:28:05	36.04007	97.506085	0
8130	97.506085-120.88260625	36.04007-53.060105	2008-02-04	18:38:47	36.04007	97.506085	0
6340	24.37652125-47.7530425	0.0-17.020035	2008-02-05	06:26:56	0.0	24.37652125	0
8131	97.506085-120.88260625	36.04007-53.060105	2008-02-07	03:48:05	36.04007	97.506085	0
1	97.506085-120.88260625	36.04007-53.060105	2008-02-07	05:31:29	36.04007	97.506085	0
4	97.506085-120.88260625	36.04007-53.060105	2008-02-07	19:01:54	36.04007	97.506085	0
2	97.506085-120.88260625	36.04007-53.060105	2008-02-04	18:06:24	36.04007	97.506085	0

Fig 3.6: Dataset after Heuristic clustering

3.3.3.2.Optimized_K-Means Function

This code anonymizes a dataset containing latitude and longitude information by applying two clustering algorithms: k-means and iterative k'-means. The iterative k'-means algorithm ensures a minimum number of trajectories per cluster for better anonymity. The resulting anonymized dataset includes the original data and additional columns for cluster assignments from both algorithms.

INPUT: Partially Anonymized Dataset

OUTPUT: Anonymized Dataset

1. Read the input dataset from a CSV file using Pandas.
2. Split the hyphenated latitude and longitude strings and convert them to float values.
3. Extract the relevant columns ('Longitude' and 'Latitude') as a NumPy array.
4. Define a function k_means that performs the standard k-means clustering algorithm on the data.

a. Initialize centroids randomly.

b. Iteratively assign data points to the closest centroids and update the centroids with the cluster means.

c. Stop when the centroids converge or the maximum number of iterations is reached.

5. Define a function optimized_k_means that performs the iterative k'-means clustering algorithm on the data.

a. Run the standard k-means clustering first.

b. Iteratively remove trajectories belonging to clusters with at least k trajectories.

c. Assign the remaining trajectories to the closest centroids.

d. Update the centroids with the new assignments.

e. Stop when the centroids converge or the maximum number of iterations is reached.

6. Run the k-means clustering algorithm on the data with a specified number of clusters.
7. Add the cluster assignments from k-means to the original DataFrame.
8. Run the optimized_k_means clustering algorithm on the data with a specified number of clusters.

9. Add the cluster assignments from optimized_k_means to the original DataFrame.
10. Save the modified DataFrame with the cluster assignments to a new CSV file.

OUTPUT:

ID	Date	Time	Latitude	Longitude	Cluster
1	2008-02-02	15:36:08	36.04007	97.506085	1
1	2008-02-06	14:17:43	36.04007	97.506085	1
1	2008-02-06	14:27:43	36.04007	97.506085	1
1	2008-02-06	14:37:43	36.04007	97.506085	1
1	2008-02-06	14:47:43	36.04007	97.506085	1
1	2008-02-06	14:57:43	36.04007	97.506085	1
1	2008-02-06	15:07:43	36.04007	97.506085	1
1	2008-02-06	14:07:43	36.04007	97.506085	1
1	2008-02-06	15:17:43	36.04007	97.506085	1

Fig 6: Anonymized Dataset after Optimized K-Means Clustering

4. Results and Discussion

4.1.Performance Metrics:

Privacy Trade-off metrics (IGPL):

Anonymous tree is searched by iteratively specializing a general value into child values. Each specialization operation splits each group containing the general value into a number of groups, one for each child value. Each specialization operation s gains some information, denoted by IG(s), and loses some privacy, denoted by PL(s).

$$IGPL(s) = \frac{IG(s)}{PL(s)+1}$$

Where IG is the Information Gain and PL is the privacy loss.

Information Gain (IG):

Information gain measures the reduction in entropy achieved by splitting the data based on a certain attribute (in this case, latitude or longitude). We can compute the information gain for each split in the decision tree, comparing the entropy of the parent node with the weighted sum of the entropies of the child nodes.

$$InfoGain(v) = E(T[v]) - \sum_{c} \frac{|T[c]|}{|T[v]|} E(T[c])$$

Where E(T[v]) is Entropy calculated by below formula

$$E(T[x]) = \frac{freq(T[x], cls)}{|T[x]|} * \log_2 \frac{freq(T[x], cls)}{|T[x]|}$$

Privacy Loss (PL):

Privacy loss measures the loss of anonymity incurred by specializing a parent node into child nodes. We can calculate the privacy loss for each specialization, comparing the anonymity levels before and after the split.

$$PL(s)=avg\{A(QID)-As(QID)\}$$

Where A(QID) and As(QID) denote the anonymity of before and after the specialization.

Reconstructor Error metrics (RE):

Reconstruction error measures the difference between the original data and the anonymized data after it has been reconstructed. Higher reconstruction error indicates higher information loss, as it implies that the anonymized data cannot accurately reconstruct the original data. Conversely, lower reconstruction error suggests lower information loss, as the anonymized data can closely approximate the original data.

$$MSE = (1/N) * \Sigma(x\_i - x\_i\_hat)^2$$

Where:

N is the total number of data points

x_i is the original data point

x_i_hat is the reconstructed data point

4.2.Result Analysis

Output Analysis of Information Loss and IGPL of existing systems

Table 4.1. Existing Methods with their information loss and IGPL

Existing Methods	Metrics
Improved Mondrian	RE=0.0001973557827121
K-Means Clustering	RE=0.0001817370943684
KNN-Cluster	RE=0.0002571819899267
Top-Down Approach	IGPL for Latitude = 0.3097 551988027 IGPL for Longitude=0.232389139918

Output Analysis Information Loss and IGPL of Proposed systems

Table 4.2. Proposed Methods with their information loss and IGPL

Proposed Model	Metrics
DGH Trees	IGPL for Latitude =3.16999766256 IGPL for Longitude=2.6747691181
Location Anonymization Framework	RE = 0.0001484011

Comparison Graph of IGPL between existing and proposed methods:

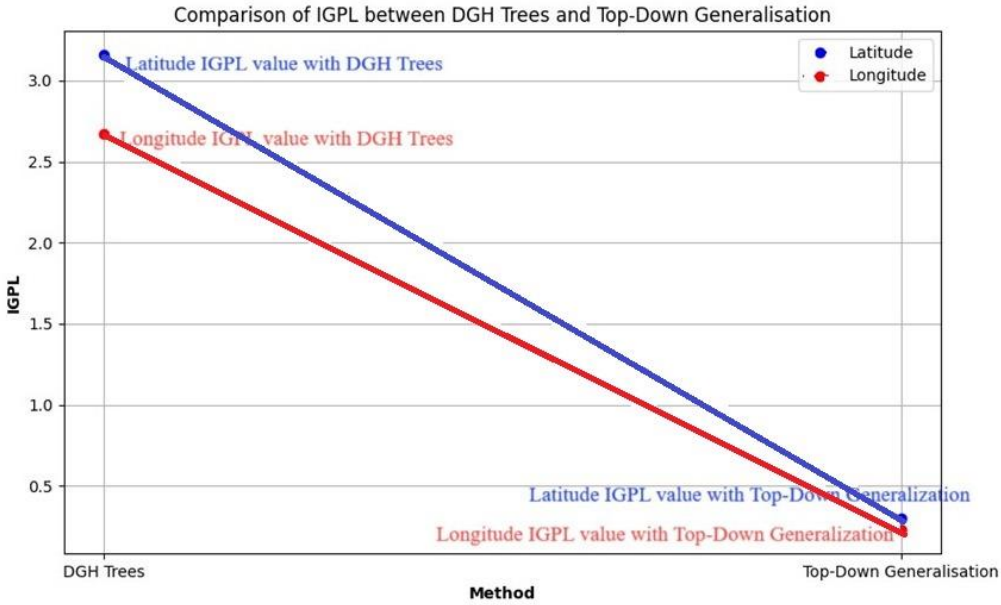


Fig 8 : Comparison graph of IGPL

The above graph revealed a notable difference in the IGPL values between DGH Trees and existing top-down generalization algorithms. Specifically, DGH Trees exhibited higher IGPL values, indicating a more effective trade-off between information gain and privacy loss. Conversely, existing top-down generalization algorithms demonstrated lower IGPL values, suggesting a lesser degree of privacy protection.

Comparison between Heuristic and Optimized K-means Clustering

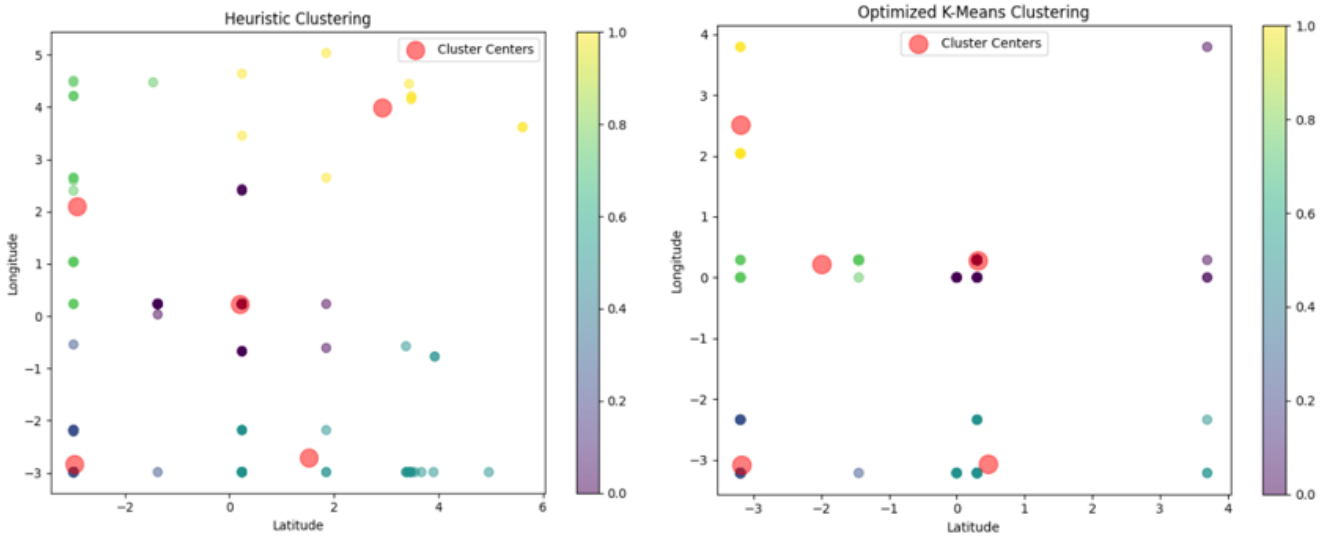


Fig 7: Comparison graph of Clustering

Silhouette Score: The Silhouette Score is a measure that evaluates the quality of clustering by comparing the intra-cluster distances (distances between points in the same cluster) and inter-cluster distances

(distances between points in different clusters). The Silhouette Score ranges from -1 to 1, where a higher score indicates better clustering.

Davies-Bouldin Index (DBI): DBI is a metric for evaluating the validity of a clustering solution. It is calculated as the average similarity of each cluster with the cluster most similar to it. The similarity is defined as the ratio between inter-cluster and intra-cluster distances. A lower DBI value indicates more compact and well-separated clusters, which is desirable.

Calinski-Harabasz Index (CHI): CHI is defined as the ratio of the between-cluster separation (BCSS) to the within-cluster normalized by their number of degrees of freedom. BCSS measures how well the clusters are separated from each other (the higher the better), while WCSS measures the compactness or cohesiveness of the clusters (the smaller the better). The Calinski-Harabasz Index is calculated as the ratio of the between-cluster variance (BCV) to the within-cluster variance (WCV), as shown in the following formula: CH = (BCV / (k-1)) / (WCV / (n-k)) where k is the number of clusters, n is the number of data points, BCV is the between-cluster variance, and WCV is the within-cluster variance. A higher Calinski-Harabasz Index indicates better clustering, as it suggests that the clusters are well-separated and compact.

Optimized K-Means Clustering Silhouette Score: 0.9667100117860429
Heuristic Clustering Silhouette Score: 0.9443618164209042
Optimized K-Means Clustering Davies-Bouldin Index: 0.4149892729368501
Heuristic Clustering Davies-Bouldin Index: 0.49208990131996877
Optimized K-Means Clustering Calinski-Harabasz Index: 8627.325711446787
Heuristic Clustering Calinski-Harabasz Index: 3515.6968191357214

Comparison Graph of information loss between existing and proposed methods

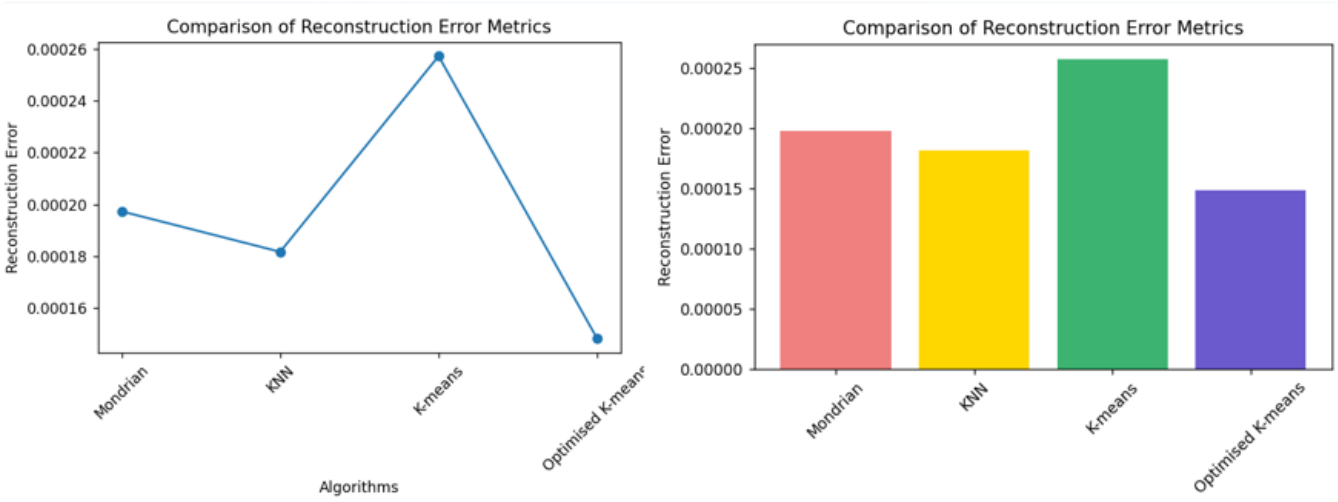


Fig 8: Comparison graph of information loss

The above graph compares the information loss of proposed and existing methods with bar plot. Lower RE represents less information loss and higher RE represents more information loss. Among all the algorithms, Optimized K-Means exhibited lower information loss, conversely existing algorithms demonstrated more information loss than the proposed one.

In conclusion, our findings provide empirical evidence that DGH Trees outperform existing top-down generalization algorithms in terms of privacy protection, as indicated by higher IGPL values and proposed

Optimized K-Means gives us the less information loss. This underscores the effectiveness of DGH Trees and Optimized K-Means in preserving privacy and reducing information loss while maintaining data utility.

5. Conclusion

In conclusion, our proposed project aims to create a machine learning framework for anonymizing location-based datasets, with a focus on balancing privacy preservation and data utility. The framework consists of three main stages: generalization using Domain Generalization Hierarchy (DGH) Trees, iterative multisequence alignment, and optimized K-Means clustering. The generalization process involves substituting specific location data with more generalized representations using DGH trees, which organize geographic entities hierarchically. Iterative multisequence alignment refines the generalized output to reduce information loss and preserve inherent patterns in the data. The optimized K-Means clustering stage partitions the partially anonymized dataset into clusters using an enhanced algorithm, ensuring privacy protection while maintaining data utility.

The project's future scope lies in refining existing techniques and exploring new methods to further improve the balance between privacy preservation and data utility. This could include incorporating additional layers of obfuscation, refining sequence alignment techniques, or optimizing the clustering process to better preserve the data's inherent patterns. Additionally, the project could be expanded to address other types of datasets beyond location-based data, ensuring a broader range of privacy-preserving solutions for various data contexts. By continuously improving the framework and expanding its applicability, the project can contribute significantly to the development of privacy-preserving machine learning techniques, ultimately benefiting both individuals and organizations in an increasingly data-driven world.

6. References

1. Bo Liu., Shuping Dang, Zihuai Lin, Jun Li, “Privacy Preserving Location Data Publishing: A Machine Learning Approach”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 33, NO. 9, SEPTEMBER 2021.
2. Anup Maurya1 ,Manuj Joshi2 , “Balancing Privacy and Utility in K-Anonymity: A Comparison of TopDown, Mondrian, and Improved”, International Journal of Intelligent Systems and Applications in Engineering,2023.
3. Dhananjay M. Kanade1 , Prof. Dr. Shirish S. Sane2, “Evaluating the Effectiveness of Clustering-Based K-Anonymity and KNN Cluster for Privacy PreservationInternational Journal of Intelligent Systems and Applications in Engineering,2023.
4. Zhenpeng Liu 1,2 , Dewei Miao 2 , Ruilin Li 2 , Yi Liu 1 and Xiaofei Li 1,*, “Cache-Based Privacy Protection Scheme for Continuous Location Query ”, Protection Scheme for Continuous Location Query. Entropy, 2023.
5. Long LI1,2 , Jianbo HUANG3 , Liang CHANG 2 , Jian WENG1 , Jia CHEN3 , Jingjing LI 1, “DPPS: A novel dual privacy-preserving scheme for enhancing query privacy in continuous location-based services” ,Research Article 2023.
6. Srinivas 1 , p. Venkata sai kumar2 , s. Siva sai krishna3 , sk. Allaudhin basha4 , t. Ganesh5, “A novel machine learning approach for privacy preserving location data publishing”, international journal 2023.

7. Mr. B. Hari babu1 , g. Sai keerthi2, “Securing location data using machine learning techniques”, journal of engineering sciences,vol 14 issue 08,2023
8. Waranya Mahanan1 · W. Art Chaovalitwongse2 ·Juggapong Natwichai3 "Data privacy preservation algorithm with k-anonymity." World Wide Web (2021) Article, 2021.
9. Sam Fletcher *, Md Zahidul Islam. "An anonymization technique using intersected decision trees” IEEE,2015
10. Jue Wang1* and Mei-Po Kwan2,3,4" Daily activity locations k-anonymity for the evaluation of disclosure risk of individual GPS datasets." 2020, International Journal of Health Geographics.IEEE, 2020.
11. jordje Slijepcevi ˇ c´ a,1,∗, Maximilian Henzl b,1,Lukas Daniel Klausner b. " k-Anonymity in practice: How generalization and suppression affect machine learning classifiers." Computers & Security,Volume 111, December 2021, 102488.
12. Feten Ben Fredja,b, Nadira Lammaria*, Isabelle Comyn-Wattiaua,c.et.al." Abstracting Anonymization Techniques: A Prerequisite for Selecting a Generalization Algorithm." 2015 19th International Conference on Knowledge Based and Intelligent Information & Engineering Systems, 2015. IEEE, 2015.
13. Anastasiia Girka a, Vagan Terziyan a, Mariia Gavriushenko a, Andrii Gontarenko bet al. " Anonymization as homeomorphic data space transformation for privacy-preserving deep learning." International Conference on Industry 4.0 and Smart Manufacturing,2012.
14. P. R. Bhaladhare and D. C. Jinwala, “Novel approaches for privacy preserving data mining in k-anonymity model,” JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 32, 63-78 (2016)vol. 32, no. 1, pp. 63–78, 2016.
15. C. C. Aggarwal, “On k-anonymity and the curse of dimensionality,” VLDB 2005 - Proc. 31st International Conference. Very Large Data Bases, vol. 2, pp. 901–909, 2005.
16. B. Sowmiya and E. Poovammal, “A Heuristic K-Anonymity Based Privacy Preserving for Student Management Hyperledger Fabric blockchain,” Wirel. Pers. Commun., vol. 127, no. 2, pp. 1359–1376, 2021.
17. D. Slijepčević, M. Henzl, L. Daniel Klausner, T. Dam, P. Kieseberg, and M. Zeppelzauer, “k-Anonymity in practice: How generalisation and suppression affect machine learning classifiers,” Comput. Secur., vol. 111, 2021
18. W. Mahanan, W. A. Chaovalitwongse, and J. Natwichai, “Data privacy preservation algorithm with k-anonymity,” World Wide Web, vol. 24, no. 5, pp. 1551–1561, 2021.
19. Y. C. Tsai, S. L. Wang, I. H. Ting, and T. P. Hong, “Flexible sensitive K-anonymization on transactions,” World Wide Web, vol. 23, no. 4,pp. 2391–2406, 2020.
20. W. Mahanan, W. A. Chaovalitwongse, and J. Natwichai, “Data anonymization: a novel optimal k-anonymity algorithm for identical generalization hierarchy data in IoT,” Serv. Oriented Comput. Appl., vol. 14, no. 2, pp. 89–100, 2020.
21. J. Wang and M. P. Kwan, “Daily activity locations kanonymity for the evaluation of disclosure risk of individual GPS datasets, ”International Journal of Health Geography, vol. 19, no. 1, pp. 1–14, 2020.
22. K. Arava and S. Lingamgunta, “Adaptive k-Anonymity Approach for Privacy Preserving in Cloud,” Arabic Journal of Science and Engineering, vol. 45, no. 4, pp. 2425–2432, 2020.
23. K. Murakami and T. Uno, “Optimization algorithm for kanonymization of datasets with low information loss,” International Journal Information Security, vol. 17, no. 6, pp. 631–644, 2018.

24. F. Ben Fredj, N. Lammari, and I. Comyn-Wattiau, “Abstracting anonymization techniques: A prerequisite for selecting a generalization algorithm,” Procedia Compututer Science, vol. 60, no. 1, pp. 206–215, 2015.
25.]B. B. Mehta and U. P. Rao, “Improved l-diversity: Scalable anonymization approach for Privacy Preserving Big Data Publishing,” Journal of King Saud University - Compututer and Information Science, vol. 34, no. 4, pp. 1423–1430, 2022.