School of Computer Science and Engineering

Nanyang Technological University

AY2024/25

AI6128 Urban Computing

Project 2

Group Members:

Kee Ming Yuan (G2304842E)

Sui Jinhong (G2302859J)

Sathananthar Suresh (G2304824C)

Robert George Susanto (G2303654A)

Content

Content 2
Contributions of Each Team Member 3
(Task 1) Data Preparation for Taxi Trajectory in Porto, Portugal 4
(Task 2) Visualization of Raw GPS Plots for the first 15 trips 5
(Task 3) Map Matching trajectory data to the road network 8
Outlier Identification........ 8
Obtaining Road Segment IDs from the GPS points 8
Limitations of the map-matching algorithm 8
(Task 4) Route Visualization 9
(Task 5) Route Analysis 10
Top 10 most traversed road segments 10
Top 10 road segments with the longest time spent 11
(Bonus Task) Case Studies 12
Outlier Identification........ 12
Local Outlier Factor (LOF) 12
The Outlier Identification method proposed by the team........ 14
Outlier removal results........ 16
Enhancement 22
Conclusion 23
References 25

Contributions of Each Team Member

Kee Ming Yuan

1. Conducted analysis of GPs points in task 2 where cases of taxi trips with outlier GPS points are observed
2. Did the entire task 6 where she investigated how LOF does not classify GPS outliers well and proceeded to think of a method to identify outliers.
3. Ran the codes derived from Suresh and Jin Hong for task 3 till task 6 with outlier GPS points removed
4. Derived the method for identifying the duplicated GPS point coordinates for task 5 part 1
5. Derived the method for identifying the top 10 longest average time spent on the road segments
6. Contributed to writing all tasks
7. Compiled codes for task 1 to 6

Sui Jinhong

1. Based on Suresh's code, made several enhancement to task 3-5
2. Contributed to the writing of all tasks

Sathananthar Suresh

1. Initiated the code generation for Tasks 1 to 5. Everything else was built upon, improved or enhanced.
2. Developed the visualization for the various tasks using an interactive map.
3. Aided in the analysis of the outliers.
4. Contributed to the report generation.

Robert George Susanto

1. Contributed to the report write up

(Task 1) Data Preparation for Taxi Trajectory in Porto, Portugal

The initial task of this project focused on preparing data for a map-matching exercise using taxi trajectories from Porto, Portugal. The road network data for Porto was obtained from OpenStreetMap and processed with tools like OSMnx to generate a navigable network covering the city's geographical boundaries [1] [2]. The trajectory data, sourced from Kaggle [3], was limited to the first 1,500 trips in the train.csv dataset, which were saved as train_1500.csv. This dataset included key attributes such as TRIP_ID, POLYLINE (a sequence of GPS coordinates), TAXI_ID, and TIMESTAMP. The POLYLINE column was particularly crucial for map-matching tasks, as it outlined the sequence of locations visited during trips.

Exploratory data analysis revealed significant variation in trip lengths, with some trips spanning over 67 km, while others were extremely short or invalid (e.g., an empty POLYLINE). This cleaned and prepared dataset provided the foundation for subsequent tasks, such as snapping GPS points to road segments and implementing map-matching algorithms. Visualizations, such as polyline length distributions, offered valuable insights into trip patterns and helped identify and address outliers in later stages of analysis.

(Task 2) Visualization of Raw GPS Plots for the first 15 trips

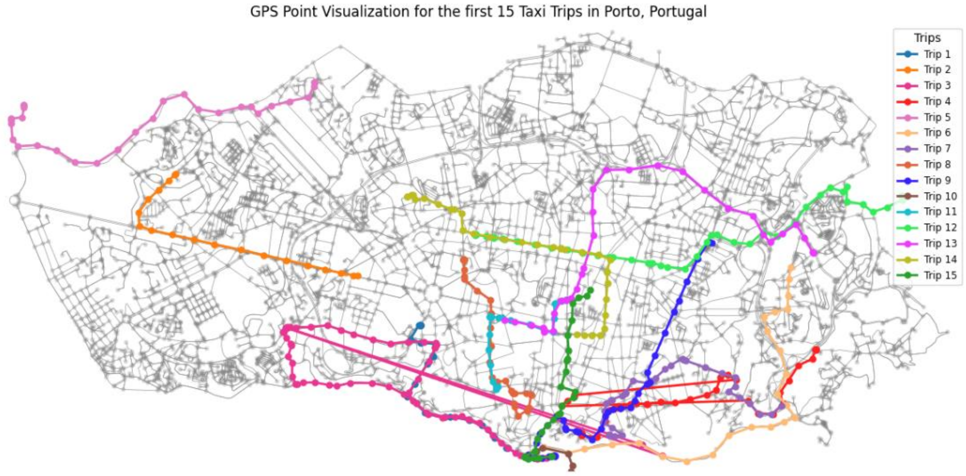


Figure 1: Visualization of the stream of GPS points of the first 15 taxi trips.

Using the OSMnx library, the visualization in Figure 1 depicted the GPS trajectories of the first 15 taxi trips in Porto, Portugal, overlaid on the road network map. Each trip was represented by a distinct colour. While most trajectories aligned well with the road network, others appeared to contain erroneous data, which we will explore and evaluate in the subsequent sections.

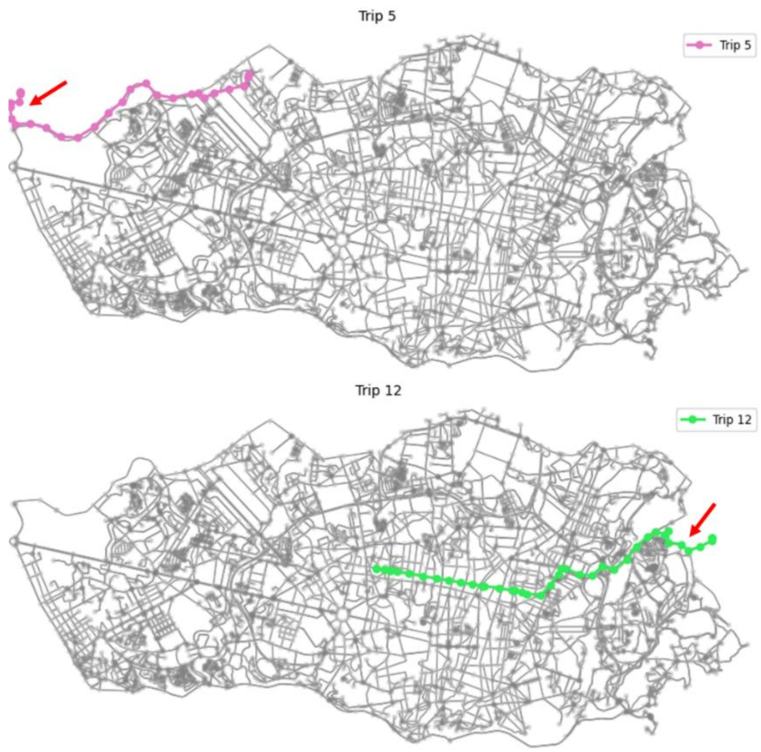


Figure 2: GPS points with no road segments under the Porto map

The OSMnx library exhibited certain limitations. For instance, while the complete set of GPS points for Trip 5 and Trip 12 were plotted, some segments of the taxi trips did not align with the visualized street network, as shown in Figure 2. This discrepancy occurred because the street network data obtained through the OSMnx library for Porto City was defined using bounding boxes, which exclude areas outside the specified geographic extent.

However, this limitation did not affect the map-matching process. Even when GPS points extended beyond the visualized map's boundaries, the OSMnx library's spatial indexing and OpenStreetMap (OSM) data were effectively utilized to determine road segment IDs, maintaining analytical precision.

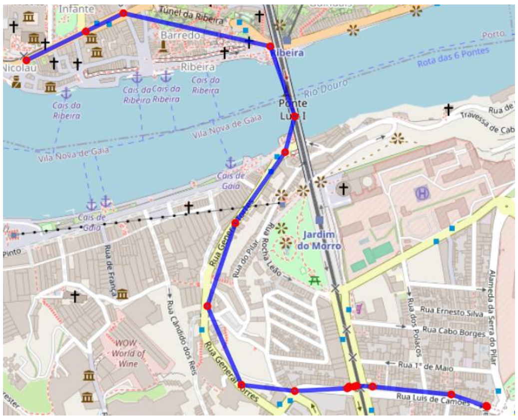


Figure 3: GPS point stream of trip 10.

Referring to Figure 1, the trajectory of trip 10 was not fully visualized. To overcome the visualization gap and offer a more comprehensive representation of taxi trips, we made use of the Folium library, as illustrated in

Figure 3. Folium allows for the creation of interactive [4], web-based maps capable of dynamically visualizing GPS points and road networks across a broader geographic extent. This approach effectively addresses the bounding box limitation inherent in OSMnx and enables a more detailed examination and interpretation of trip trajectories.

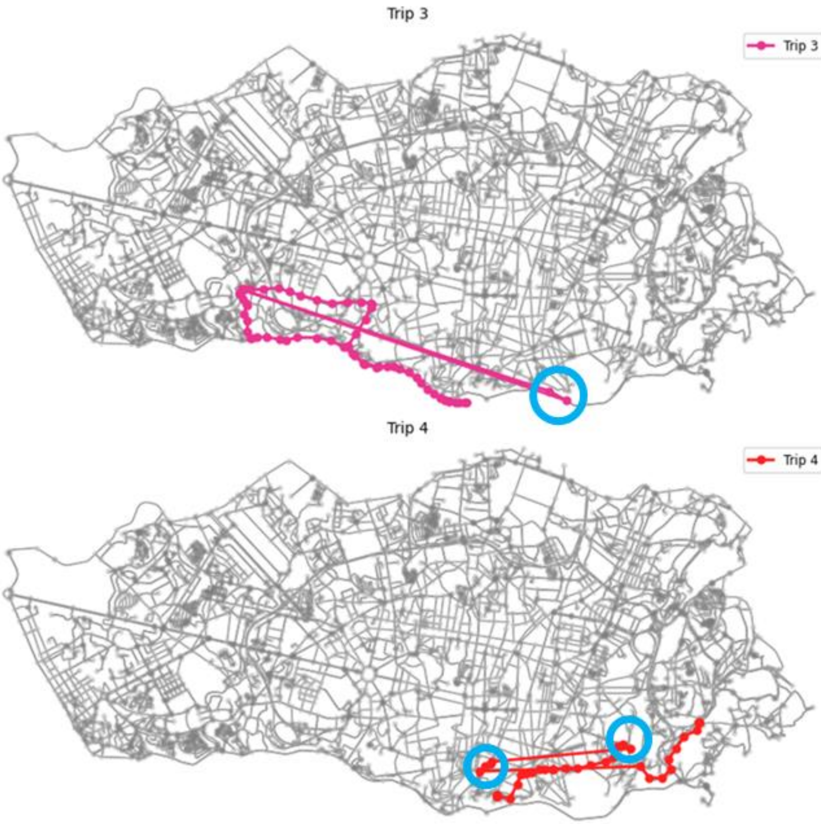


Figure 4: GPS point streams for trip 3 and 4.

Another issue identified was the presence of GPS outliers in some taxi trips, where sudden and big deviations of GPS location from its previous location can be seen, as circled in Figure 4Correcting such erroneous data points was essential because they could result in inaccurate snapping or map matching by incorrectly aligning GPS points to unrelated road segments.

(Task 3) Map Matching trajectory data to the road network

Outlier Identification

As outlined in Task 2, GPS outliers were identified as a significant issue in the dataset, affecting numerous taxi trips. To address this, the team devised a method for handling outliers and erroneous GPS data. Further details about this approach, as well as the challenges where the map-matching algorithm performed suboptimally in the presence of GPS outliers, will be elaborated in Task 6. Task 3 will therefore be conducted without the removal of GPS outliers.

Obtaining Road Segment IDs from the GPS points

The longitude and latitude of each GPS point were converted into point geometries, with the dataset assigned the Coordinate Reference System (CRS) EPSG:4326, which aligns with the global WGS 84 (World Geodetic System 1984) standard [5]. Subsequently, the GeoDataFrame was reprojected to CRS EPSG:32629 to facilitate proximity calculations. For each GPS point, the closest road segment was identified using Euclidean distance, and the corresponding osmid was stored in new columns. To optimize the process, the performance of the cKDTree algorithm was compared with standard Euclidean distance calculations. The final output of task 3, containing the road segments of all 1497 trips, can be found in **'trip_segment_euclidean_pertrip.csv'**.

Limitations of the map-matching algorithm

The map-matching algorithm demonstrated some limitations. Since it matched GPS points to the nearest road segment, gaps occasionally appeared in the reconstructed route. These gaps typically arose at junctions or road segment transitions without capturing GPS points. To mitigate such limitations, more robust open-source tools like FastMapMatching or MappyMatch could enhance map-matching accuracy [6] [7].

(Task 4) Route Visualization

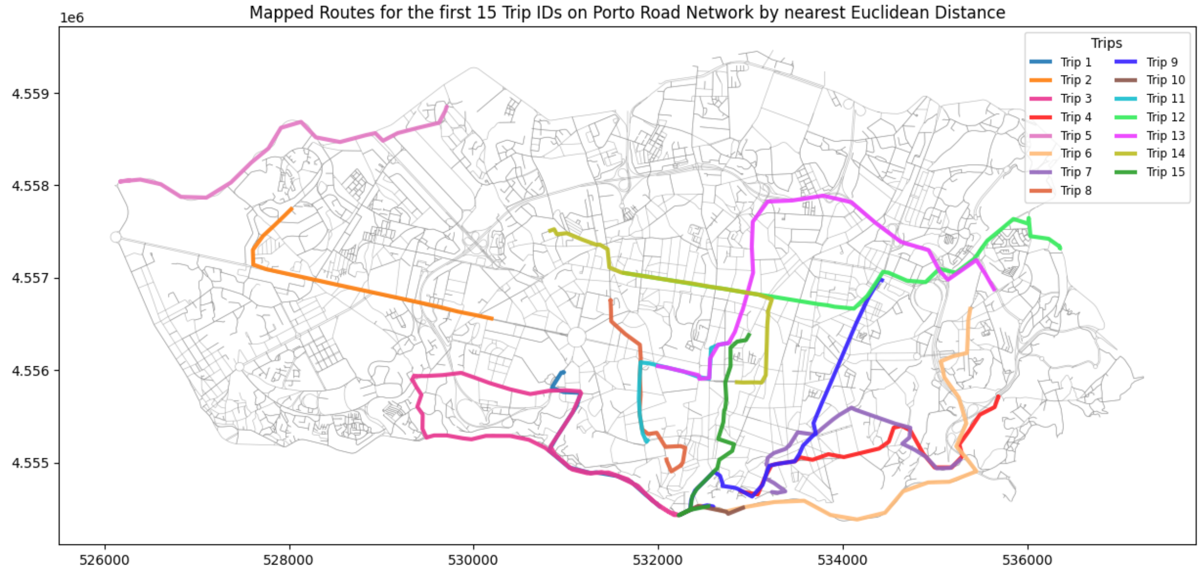


Figure 5: Road segment traversed by the first 15 taxi trips using nearest Euclidean distance

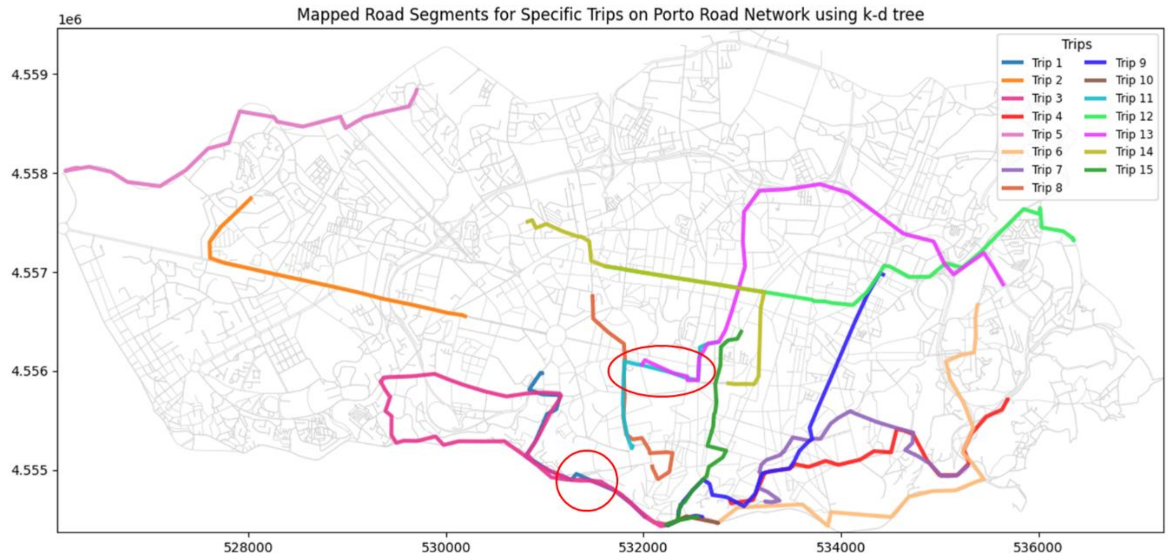


Figure 6: Road segment traversed by the first 15 taxi trips while using k-d tree to identify the road segment IDs

As illustrated in Figure 5 & Figure 6, the team optimized Task 3 by comparing two methods for identifying the nearest road segments to GPS points: the k-d tree approach (Figure 6) and the Euclidean distance method (Figure 5). While there were slight differences in the nearest road segments identified, as circled in Figure 6, the results from both methods were nearly identical. A key observation was that the k-d tree method significantly reduced computation time (around 50%) compared to the Euclidean distance approach. However, the results derived using the shortest Euclidean distance method were utilized for consistency for subsequent tasks.

(Task 5) Route Analysis

Top 10 most traversed road segments

Since the GPS data was recorded at 15-second intervals, the same road segment often appeared consecutively when the taxi required more than 15 seconds to traverse that road. This can lead to inaccurate identification of the top 10 most traversed roads. To address this, consecutive duplicate road segments were removed, retaining only one instance each. The team also recognised that some trips comprise U-turns or round trips where the taxi covered the same road segment more than once. In this case, the repeated road segments are not removed. The algorithm can be described as follows:

1. **Sequential Comparison of Road Segment IDs**: Road segment IDs were compared sequentially for each trip. If a sequence of road segment IDs (e.g., IDs 1-10) were identical, only one instance of the road segment was recorded.
2. **Handling Repeated Road Segments in Round Trips**: If a road segment (e.g., IDs 1-10) appeared again later in the trip (e.g., IDs 15-17), the repeated segment was recorded again but as a single instance for each occurrence.

Using this algorithm, the team identified the top 10 most traversed road segment IDs presented in Table 1, with a corresponding visualization in Figure 7.

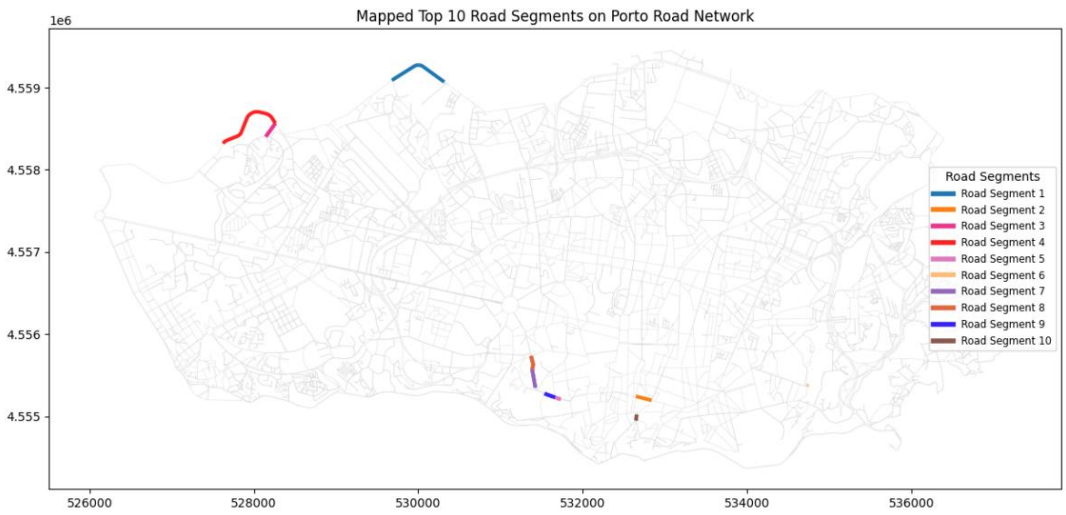


Figure 7: Top 10 most traversed road segments

Table 1: Top 10 most traversed road segment IDs

Index	Road Segment OSMID	Count	Index	Road Segment OSMID	Count
1	[1016542178]	201	6	[36748084]	128
2	[434897752, 1016528107, 1226233909]	176	7	[434860177, 1174835646]	122
3	[838000172]	170	8	[287648900]	118
4	[190128487]	147	9	[398035553, 453763721, 663351958, 1171297150]	117
5	[715176795, 1184851191]	129	10	[859507142]	117

Top 10 road segments with the longest time spent

The Kaggle competition dataset specified that each data point was collected at 15-second intervals. With this, the team calculated the average time spent on each road segment without removing any duplicated road segments with the following formula:

$$\text{Total seconds on the road segment} = \left(\frac{\text{Sum of number of times the road segment appeared in all trips} - \text{Number of unique trips}}{\text{Number of unique trips}}\right) \times 15$$

Given that the taxi rarely returns to the same road segment within a single trip, the assumption was that each road segment ID appeared only once per trip. Thus, the number of unique trips was subtracted from the total occurrences of the road segment across all trips as the first GPS point is recorded at the start of the trip where the taxi is not moving. The resulting seconds were converted into minutes and seconds for better interpretability.

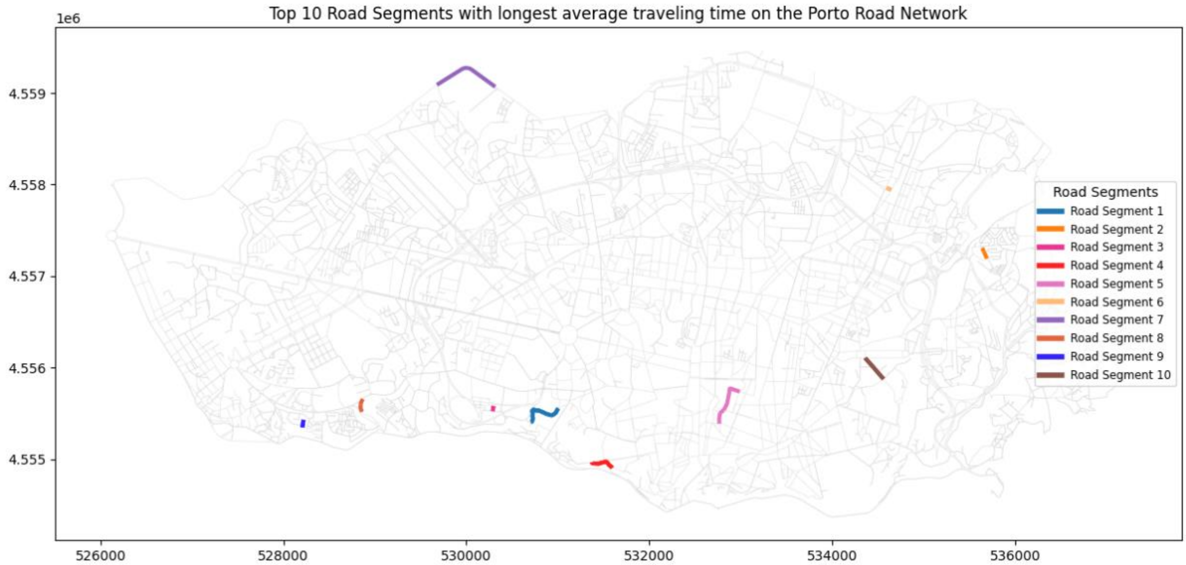


Figure 8: Top 10 road segments with the longest average time spent.

Table 2: Top 10 road segments with the longest average time spent across all trip IDs

Index	Road Segment OSMID	Average Time	Index	Road Segment OSMID	Average Time
1	[128362570, 39895274, 481849357, 480494958, 481849358]	13 minutes, 15 seconds	6	[1308709259]	6 minutes, 0 seconds
2	[134085457]	12 minutes, 15 seconds	7	[39977889]	5 minutes, 7 seconds
3	[39911797]	7 minutes, 45 seconds	8	[1172489853]	5 minutes, 7 seconds
4	[39678582]	6 minutes, 52 seconds	9	[41182338]	4 minutes, 37 seconds
5	[8367104]	6 minutes, 30 seconds	10	[34952929]	4 minutes, 30 seconds

The top 10 road segments with the longest average time spent on them, based on the analysis of 1,497 trips, are presented in Table 2 and Figure 8 respectively above.

(Bonus Task) Case Studies

As previously discussed in the report, erroneous data points or outliers were observed within the dataset. These data points exhibited irregular patterns, such as abrupt deviations or loops, and were frequently encountered in the first 1,500 trips. Such outliers are likely the result of inaccurate or weak GPS signals, which can significantly affect the accuracy of subsequent analytical tasks.

The potential causes for these erroneous GPS signals include physical obstructions between the taxi and the satellite, such as when the taxi is traveling through tunnels, near tall buildings, or in areas with dense infrastructure. These factors disrupt signal transmission and lead to inaccurate location readings, underscoring the importance of addressing these outliers to ensure data reliability.

Outlier Identification

Local Outlier Factor (LOF)

The team employed the Local Outlier Factor (LOF) algorithm to detect outliers [8], where k represents the nearest neighbors considered for each point. The LOF is calculated using the following formula:

$$LOF = \frac{\sum_{o \in N_k(p)} \frac{lrd_k(o)}{lrd_k(p)}}{k}$$

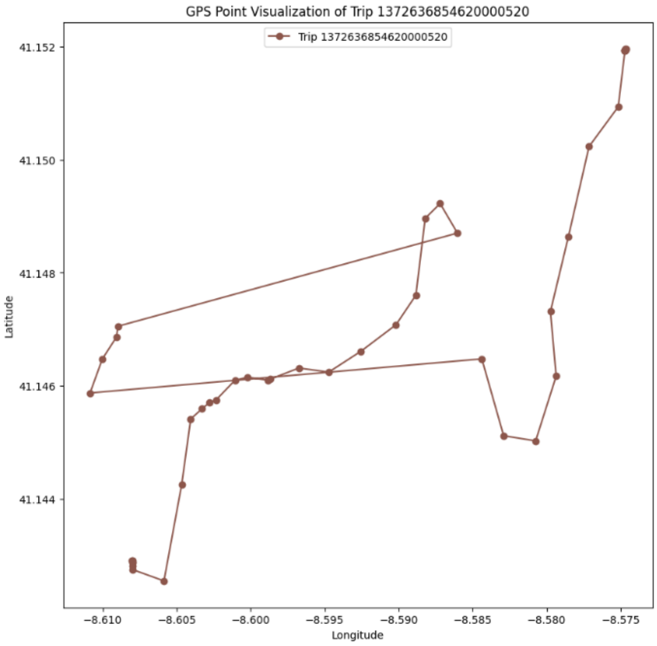


Figure 9: Zoomed in GPS points of trip 4.

By plotting the first 100 trips, the team observed that the maximum number of outliers clustering closely together in a single trip was four, which occurred in Trip 4, as shown in Figure 9 above. To reliably distinguish outliers from non-outliers, it was determined that the minimum k-value should be at least twice the number of clustered outliers, requiring a minimum k-value of 8.

To adopt a more conservative approach and enhance the algorithm's sensitivity to outliers, the team decided to increase the k-value to 10. This adjustment ensures a more robust identification of outliers, reducing the likelihood of misclassification.

Nevertheless, using LOF to identify outliers does need to lead to the GPS points being accurately identified. Looking at the Figure 10 below, for the case of trip 65, the GPS point outliers occurred at the middle of the trip and the outliers happen to be close to the correct GPS points at the start of the trip:

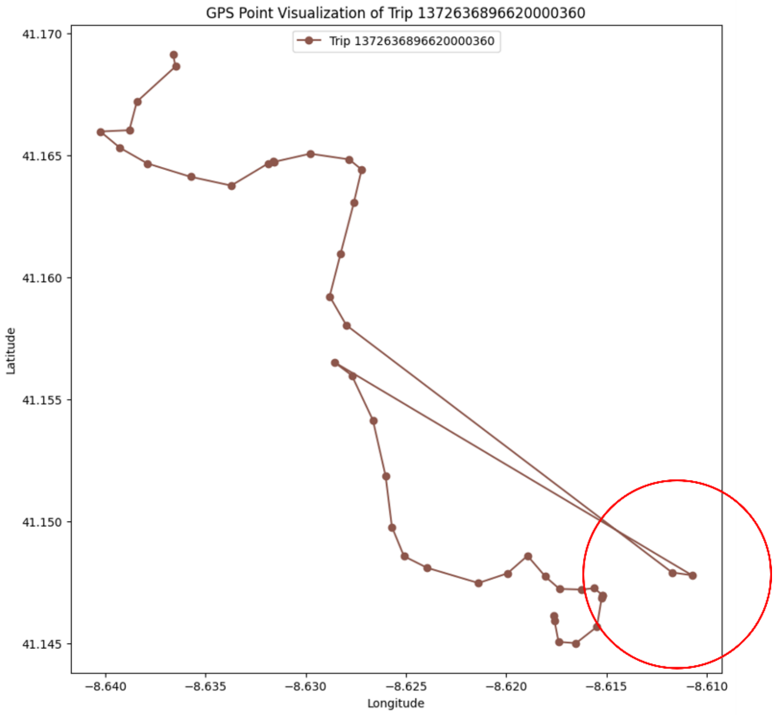


Figure 10: LOF is not good at identifying outliers for cases where outliers are close to GPS points collected during different part of the taxi journey.

As illustrated in Figure 10 above, the LOF algorithm does not account for the sequence that the GPS data is collected, where each GPS point should logically be in proximity to its preceding point. This limitation resulted in some outliers being assigned LOF values smaller than 1, incorrectly classifying them as non-outliers in certain cases.

Due to this inherent limitation, the team deemed LOF unsuitable for identifying outliers in the dataset containing 1,497 trips. The lack of consideration for the temporal or spatial sequence of GPS points undermines the algorithm's ability to effectively detect irregular patterns in this context.

The Outlier Identification method proposed by the team

Given the following considerations:

- The official express speed limit on Porto expressways is 100 km/h [9].
- GPS points are recorded at 15-second intervals.

Assuming adherence to the speed limit, the maximum distance taxis can travel in a 15-second interval would be 0.417km. For cases involving potential speeding, a larger radius around each GPS point may need to be considered for outlier detection using LOF. The areas around correct GPS points and potential outlier GPS points were examined to determine an appropriate radius. This analysis ensures that the chosen radius effectively encompasses reasonable variations in GPS data while still identifying outliers resulting from erroneous or anomalous signals.

From the observation of the trips with outliers in the first 100 trips in the dataset, trip 4 of ID 1372636854620000520 had the outlier points closest to its preceding GPS point, which was not an outlier. Hence, trip 4 was used as the benchmark for determining the best radius for the dataset. Two radius values were tested, as shown below: 1.25km and 0.625km.

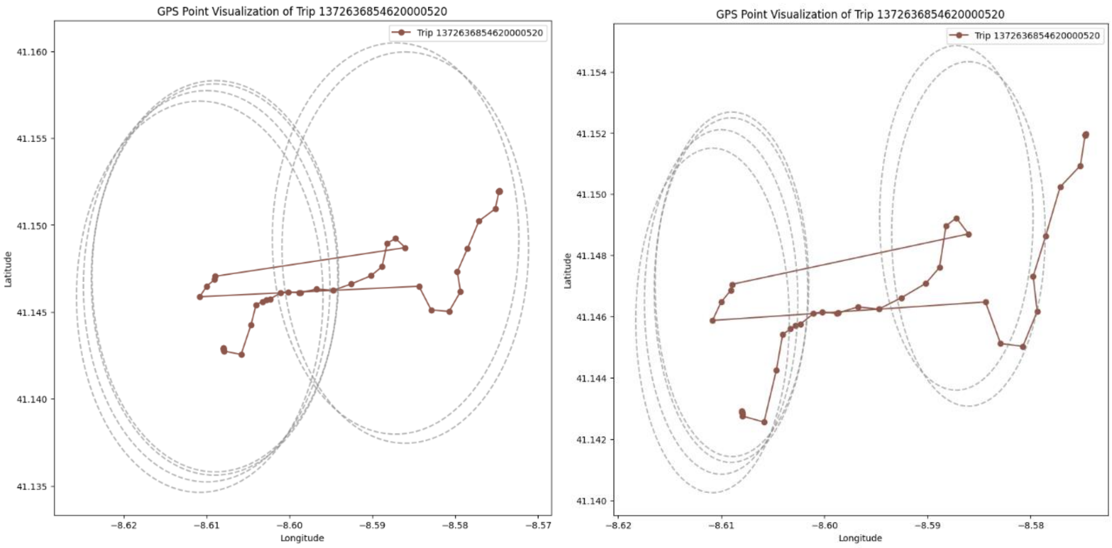


Figure 11: Adjustment of radius for calculating LOF. (Left) Diameter = 1.25km, 3 times the speed limit of Porto (Right) Diameter = 0.625km, 1.5 times the speed limit of Porto.

Based on the analysis in Figure 11, a diameter equivalent to 1.5 times the speed limit of Porto, which is a distance of 0.625 km travelled per 15 seconds, was determined to be optimal for outlier detection. Using a larger radius, such as 3 times the speed limit (1.25 km), can cause significant overlap between areas around correct GPS points and erroneous ones, making it difficult to identify outliers accurately. Consequently, a diameter of 0.625 km was adopted for this method.

The team developed a step-by-step method for detecting GPS outliers in the dataset. Hyperparameters used in this process are **bolded** to indicate their adjustability:

- Create a new column in the dataset called 'Is it an outlier' with all rows defaulted to 'Not outlier'.
- For each trip in the dataset:
    - Calculate the geodesic distance between adjacent GPS points and store the value in a new column named 'Adjacent Distance'. For example, the 'Adjacent Distance' for the 2nd GPS point records the geodesic distance between points 1 and 2. For the first GPS point in the trip, the 'Adjacent Distance' is set to 0.
    - Identify any GPS points with an 'Adjacent Distance' greater than or equal to **0.625 km**.
- Detect subsequent outlier points:

- For each identified outlier point, examine the next **9 points** to check if they have an 'Adjacent Distance' of at least 0.625 km.
    i. For example, point 4 had aa 'Adjacent Distance' of more than or equal to 0.625km geodesic distance from its previous point. We would look at the next **9 points**, starting at point 13, and see if there were any 'Adjacent Distance' of more than or equal to 0.625km geodesic distance.
        1. If there was, for example, point 11 and point 12 had a geodesic distance of more than 0.625km; we would update the value of 'Is it an outlier' for point 4 to point 11 as 'Outlier'.

For the context of this project, this approach successfully identified GPS point outliers, enabling their removal before applying the map-matching algorithm. This method was deemed effective in identifying and addressing GPS outliers within the dataset, as shown in the subsequent sections.

The identified outliers can then be removed, allowing the map-matching algorithm to more reliably associate road segments with valid GPS data, enhancing the quality and accuracy of derived taxi trip trajectories. The following considerations were noted:

- Given that GPS points were collected at short 15-second intervals, taxis likely remain on the same road segment before and after the outliers.
- Even if this is not the case, the taxis would likely still be on nearby road segments, ensuring minimal impact on the accuracy of the map-matching results.

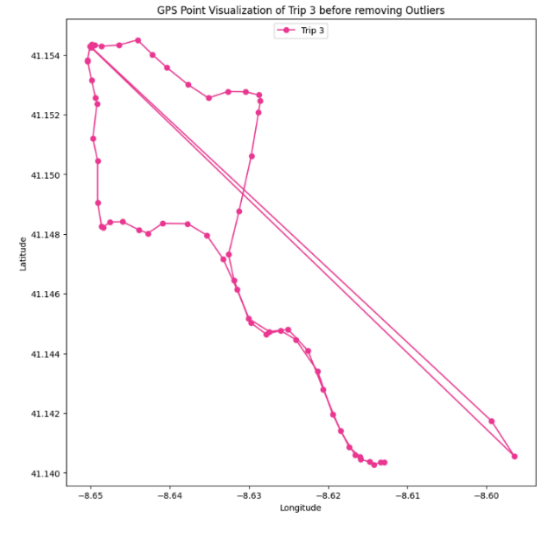
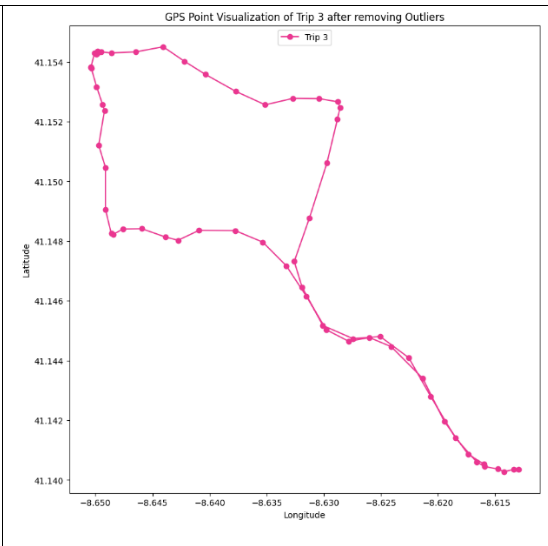
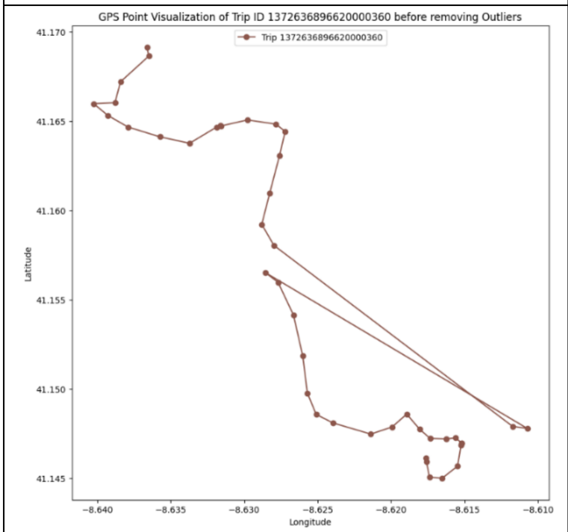
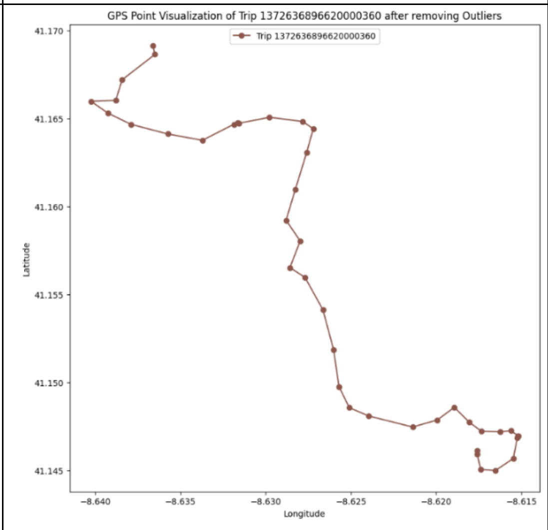
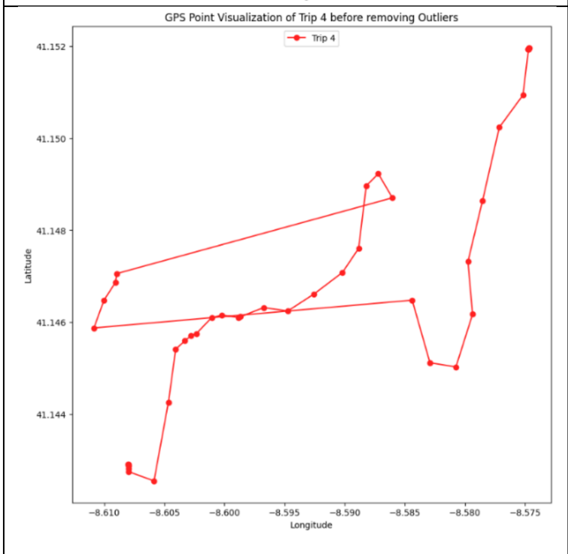
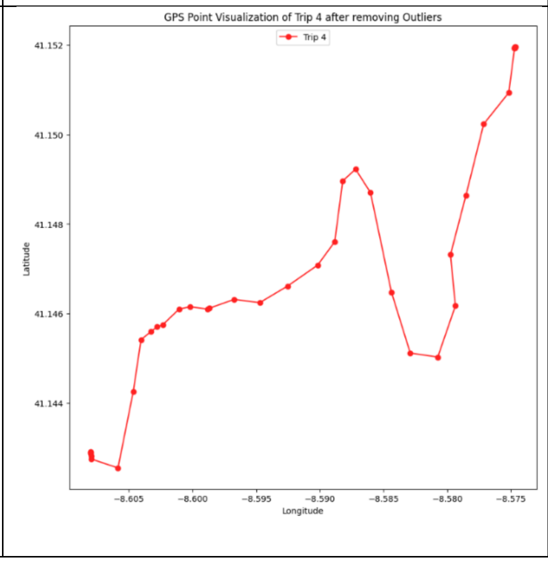
Outlier removal results

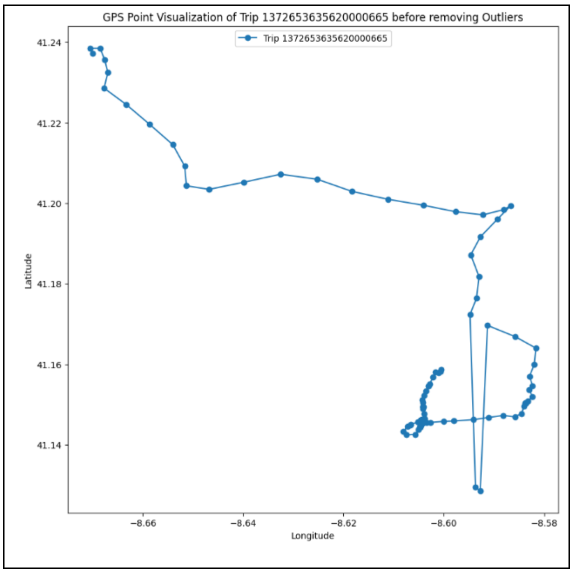
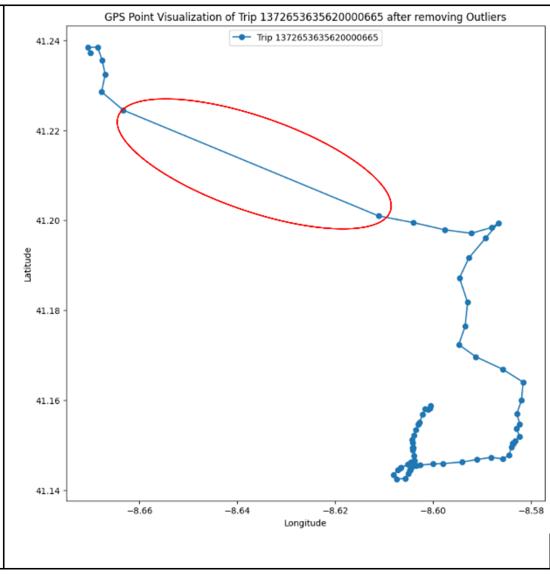
Using the proposed outlier detection method, the team identified that 5.54% of the 1,497 trips contained outlier points. Of the total 71,697 GPS points across these trips, 225 were classified as outliers.

This result highlights the prevalence of GPS anomalies within the dataset and underscores the importance of handling these outliers to maintain the accuracy of subsequent analyses, such as map matching and route reconstruction.

Table 3: Before and after removing outliers from the GPS points of the trips

Before removing Outliers	After removing Outliers	Outliers Removed

<table>
<tr>
<td>
</td>
<td>
</td>
<td>2</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
<td>2</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
<td>4</td>
</tr>
</table>

		11

As shown Table 3 above, removing outlier GPS points did result in some loss of information, especially in trips with a higher number of outliers. However, even after these points are excluded, the remaining GPS data still effectively represented the overall trajectory of the taxi routes. This suggests that the outlier removal process preserves the integrity of the routes despite discarding erroneous data points.

Examining the second row of Table 3, the proposed outlier detection technique accurately identified GPS outliers that were spatially close to other GPS points recorded at different parts of the trip. This was achieved by considering the geodesic distances and the sequential order of the GPS data.

The algorithm also proved effective in identifying trips with a significant number of outliers, such as Trip 4, which exhibited a large cluster of erroneous data points. Upon further analysis of the street map for Trip 4, it was noted that one possible cause of these outliers could be the presence of metal train tracks in the area, as illustrated in Figure 12 below. Such infrastructure could interfere with GPS signals, leading to inaccuracies in the recorded locations.

Figure 12: Street map of trip 4 zoomed in to the outlier portion.

Metallic materials could cause total reflection of GPS signals [10], which may explain the occurrence of outliers in Trip 4. This interference could distort the GPS readings, leading to erroneous data points.

In the fourth row of Table 3 above, the GPS points for trip ID 1372653635620000665 were analyzed. This trip had the highest number of GPS points removed, with a total of 9 points identified as outliers. These points were highlighted in the red-circled area on the graph in the table.

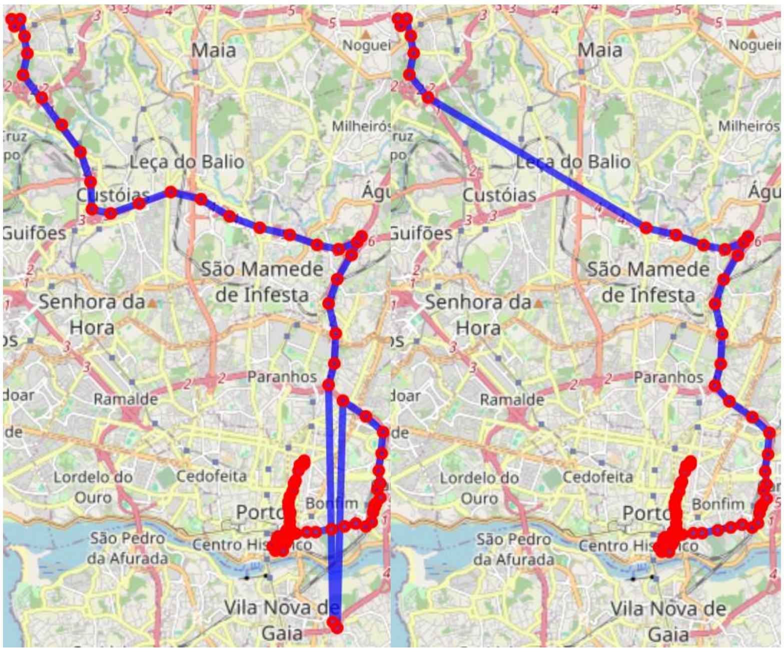


Figure 13: Street view of the GPS points of trip 1372653635620000665.

Upon closer inspection, as illustrated in figure 13 above, the 9 removed GPS points were found to lie directly on the road and were inaccurately classified as outliers. This misclassification indicates a limitation of the outlier detection algorithm in certain scenarios, particularly when legitimate GPS points share similar characteristics with identified outliers. Adjustments to the algorithm or additional validation criteria would be necessary to improve accuracy in such cases.

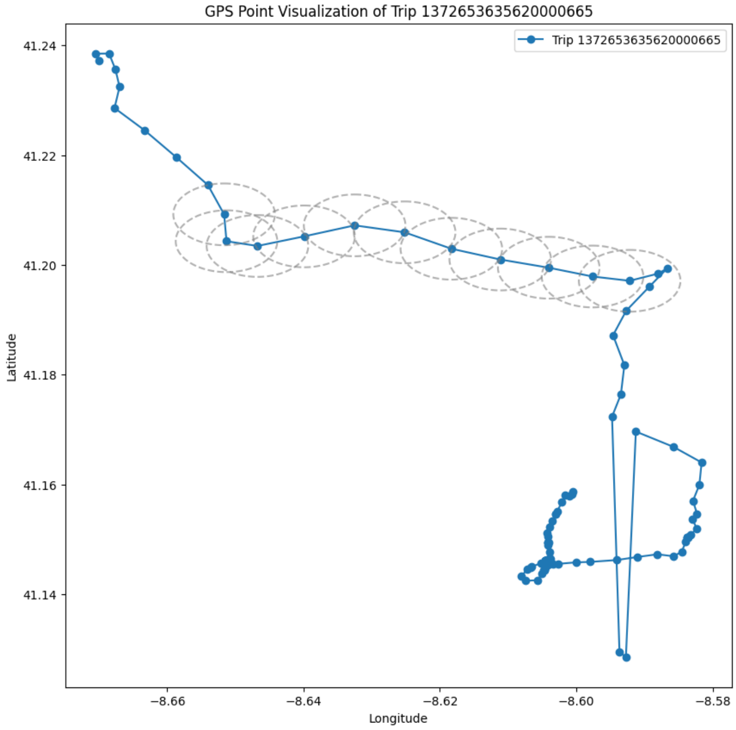


Figure 14: The 9 falsely identified outliers were outside the 0.625km distance from each other.

In figure 14 above, while the 9 GPS points did not appear to be outliers, they exceeded the 0.625 km geodesic distance diameter, which was the threshold used for identifying outliers. Each GPS point lay outside the diameter of the circle corresponding to another GPS point, with the circle's diameter being 0.625 km. This explains how these points were identified as outliers.

This analysis indicated that the algorithm proposed by the team did not effectively identify outliers for taxis speeding in Porto, where they were traveling at speeds of 1.5 times or more the speed limit (equivalent to covering 0.625 km in 15 seconds). However, taxis typically traveled at such high speeds on relatively straight roads. As a result, the overall shape of the taxi's trajectory for trip ID 1372653635620000665 was preserved even after the removal of the 9 GPS points. This suggested that the most probable road segments the taxi traveled on during map matching would not deviate significantly from the actual road segments it traversed.

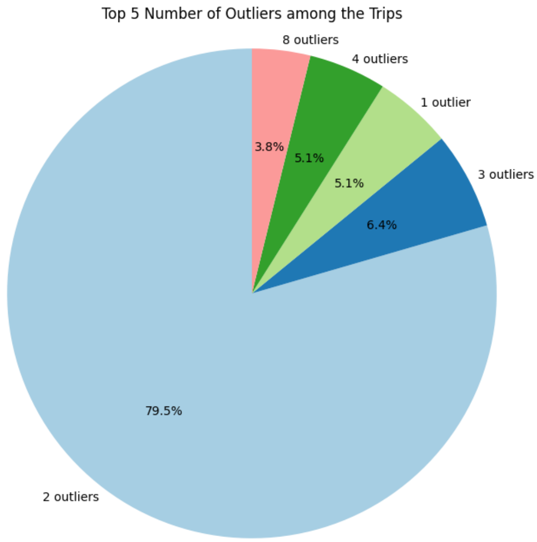


Figure 15: Distribution of the number of outliers for the trips with outliers in the 1497 trip dataset.

From figure 15 above, it was observed that most trips with outliers had only 2 outliers. As shown in the first two rows of Table 3, the proposed algorithm effectively identified outliers for trips with 2 outliers, accounting for 79.5% of the trips containing outliers. Therefore, it was concluded that the algorithm performed well in identifying outliers for this particular dataset.

Despite occasional instances of misidentifying GPS points as outliers, the analysis showed that the algorithm preserved the essential structure of the trajectories. This preservation was crucial for downstream applications such as map matching, route optimization, or traffic flow analysis. By balancing outlier removal with trajectory retention, the algorithm ensured that key features, such as road curvature and segment connectivity, remained largely intact.

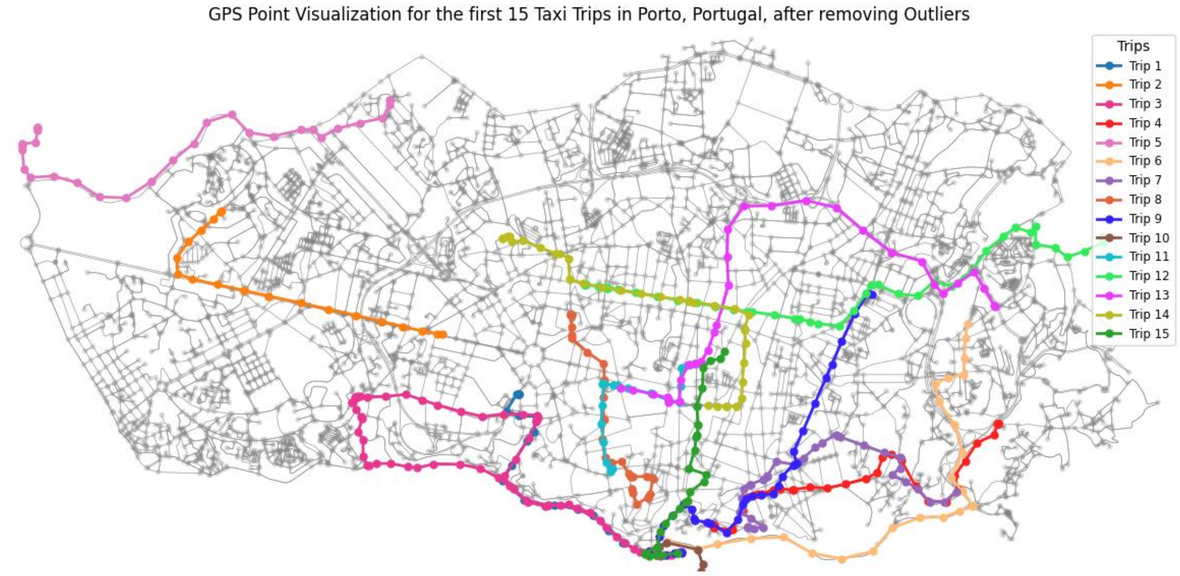


Figure 16: GPS points of the first 15 trips after the removal of outliers.

Figure 16 displayed the GPS visualization of the first 15 trips after the removal of outliers.

Enhancement

Below are some of our proposed enhancements to improve the map-matching algorithm:

Similarity in difference of adjacent distance

The difference in the adjacent distances of each point was considered to address issues such as those in trip ID 1372653635620000665, where 9 out of 11 points were incorrectly identified as outliers. An additional algorithm was proposed to refine the identification of outliers from the previously implemented algorithm (with adjustable hyperparameter values **bolded** below):

- A new column, "Difference in Adjacent Distance," was created from the "Adjacent Distance" column. For the first point in the trip, the value of "Difference in Adjacent Distance" was set to 0. For example, the "Difference in Adjacent Distance" for the second row was the absolute difference between the "Adjacent Distance" values of the first and second rows.
- For each trip with outliers identified by the earlier algorithm, an additional check was performed. If the next **9 points** from the initially flagged outlier were within a specific geodesic distance threshold such as **0.2 km**, representing half the distance allowed by the speed limit in Porto over 15 seconds, those points were marked as "False Outliers," meaning they were not considered outliers.
- If the distance exceeded 0.2 km, the points were marked as "Real Outliers" and considered valid outliers.

This refined approach aimed to account for taxis genuinely speeding, thereby improving the accuracy of outlier identification.

Multi-modal data collection

The algorithm's performance was further enhanced by integrating GPS data with other data modalities, such as road types, traffic conditions, and historical driving patterns, to provide additional context. For example, combining speed profiles with map data could refine the definition of outliers, especially in cases where taxis exceeded speed limits but followed known paths. This hybrid approach mitigated false positives and improved the overall accuracy of outlier detection.

After the outliers are identified

For road gaps created by removing outliers, existing map-matching algorithms were applied to map the shortest route between the points before and after the outlier segment. This approach was consistent with common routing systems used in driving applications to recommend optimal paths for taxi drivers to reach their destinations [11].

Conclusion

In this report, we explored taxi trajectories in Porto, Portugal, tackling challenges in GPS data processing, map-matching, and outlier detection. Through our systematic approach, we visualized raw GPS points, identified common routes, and analyzed trajectory patterns to gain a deeper understanding of the urban taxi movement.

Some of our key insights are as follows:

- **Map-matching and Route Analysis:** We successfully matched road segment IDs to GPS points by applying techniques such as Euclidean distance and k-d trees. The k-d tree method proved particularly efficient. Additionally, we analyzed the top 10 most traversed road segments and those with the longest average travel times, providing valuable insights into traffic patterns and bottlenecks.
- **Visualization Enhancements:** We used interactive maps, including Folium, to create dynamic visualizations of taxi routes. These maps highlighted areas of potential GPS errors and enabled us to propose solutions for improving trajectory mapping.
- **Outlier Detection and Removal:** We developed a custom algorithm to detect and remove GPS outliers, identifying 5.54% of points as outliers. While some inaccuracies occurred in cases involving speeding taxis, our approach preserved the overall structure of the trajectories, ensuring that key analytical outcomes were not compromised.

- **Algorithmic Improvements:** We proposed enhancements such as incorporating multimodal data and refining adjacent distance calculations. These ideas will improve outlier detection and map-matching accuracy.

Final Thoughts

This project highlighted the prevalence of erroneous GPS data and its downstream effects. To address these errors, several methods for identifying outliers were proposed. However, no single approach was universally suitable for all use cases, requiring analysts to adopt strategies tailored to the specific context of the data and project. An alternative approach involved identifying outliers and imputing them with predicted GPS data.

From the findings discussed in the previous sections, the application of GPS data was demonstrated in various tasks, such as traffic monitoring and map matching, which could significantly impact other use cases, including route planning and the enhancement of urban infrastructure.

Further research was suggested to focus on accurately identifying erroneous GPS data, aiming to create a more reliable data pool for critical applications.

References

[1] K. Curran, J. Crumlish, and G. Fisher, "OpenStreetMap," Int. J. Interact. Commun. Syst. Technol., pp. 1493–1498, 2012.

[2] "OSMnx paper," *Geoff Boeing*, 01-May-2024. [Online]. Available: https://geoffboeing.com/publications/osmnx-paper/. [Accessed: 24-Nov-2024].

[3] *Kaggle.com*. [Online]. Available: https://www.kaggle.com/datasets/kan/porto-taxi trajectories. [Accessed: 24-Nov-2024].

[4] "Folium — folium 0.18.0 documentation," Github.io. [Online]. Available: https://python-visualization.github.io/folium/. [Accessed: 24-Nov-2024].

[5] GISGeography, "World Geodetic System (WGS84)," *GIS Geography*, 06-Oct-2015. [Online]. Available: https://gisgeography.com/wgs84-world-geodetic-system/. [Accessed: 24-Nov-2024].

[6] C. Yang, fmm: Fast map matching, an open source framework in C++

[7] mappymatch: Pure-python package for map matching

[8] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," SIGMOD Rec., vol. 29, no. 2, pp. 93–104, 2000.

[9] "Portugal traffic guide," vintrica.com, 15-Apr-2024. [Online]. Available: https://www.vintrica.com/en/blog/trafficguide/portugal/?srsltid=AfmBOor4kmR3HKoBfCtz3l_SAghIRV26sMjGRus_p_Jg8HFjfBgyRC-8. [Accessed: 23-Nov-2024].

[10] T.-H. Yi, H.-N. Li, and M. Gu, "Effect of different construction materials on propagation of GPS monitoring signals," Measurement (Lond.), vol. 45, no. 5, pp. 1126–1139, 2012.

[11] R. Simmons, B. Browning, Y. Zhang, and V. Sadekar, "Learning to predict driver route and destination intent," in *2006 IEEE Intelligent Transportation Systems Conference*, 2006.