UTRECHT UNIVERSITY
Graduate School of Natural Sciences

Applied Data Science master thesis

Chaos amidst the Flock: Clustering Oil Barrel Trajectories

First examiner:
Tim Ophelders

Candidate:
Niem Schneider

Second examiner:
Bas Altena

In cooperation with:
Space Accountants

June 27, 2024

Abstract

This study explores the effectiveness of various clustering algorithms in reducing data variability in maritime oil barrel pollution analysis. The research investigates four clustering methods - *k*-Means, Agglomerative, HDBSCAN, and OPTICS - applied to simulated trajectories of oil barrels. The focus is on assessing the reduction in data variability using standard deviation and Mean Absolute Deviation (MAD) as reduction rates. The findings demonstrate that density-based clustering methods, particularly OPTICS, significantly reduce data variability by categorizing noise effectively. However, this approach may not be suitable for applications requiring the inclusion of all trajectories, such as identifying offending ships. In these scenarios, distance-based methods perform better but offer minimal data reduction. These results underscore the importance of selecting appropriate clustering methods based on specific requirements. The broader perspective suggests that while clustering can enhance data analysis efficiency, careful consideration of the trade-offs between data reduction and information retention is essential for reliable maritime pollution tracking.

Key words: Cluster Algorithms, Data Reduction, Trajectory Analysis, Oil Barrel Pollution, Distance-based Clustering, Density-based Clustering.

Contents

1. Introduction

1.1 Motivation and Context

On the 25th and 26th of February 2023, eleven oil barrels were found along the coast of Vlieland, Terschelling and Ameland, part of the Wadden Islands in the Netherlands. The most likely source is a single ship, navigating through the North Sea and discarding the barrels after using the oil, for instance as lubrication for the engine. Currently, the local municipalities are responsible for cleaning up the mess, but they lack the resources to investigate who committed this act. The littering of oil barrels into the North Sea impacts the quality of marine life, contributes to waste on the shoreline and imposes unforeseen expenditures on municipalities (Bascom, 1974). In order to change this type of behaviour, it is crucial to track down the vessel responsible for the environmental pollution.

1.2 Literature Overview

One way to identify the responsible vessel is by simulating the possible trajectories of the oil barrels using ocean models. These models are based on observations made by weather stations and technological institutes and simulate natural phenomena such as waves, wind, temperature fluctuations, tides and more (BRYAN, 1969). They enable the simulation of the possible paths the oil barrels might have taken to reach their locations. Ideally, this process can pinpoint a single ship as the origin of the trajectories of the barrels. Once identified, this vessel can be considered to be the offending ship, and municipalities can take further action.

The simulation results in a complex web of spatio-temporal patterns, also defined as a flock (Gudmundsson & van Kreveld, 2006). Moving objects can exhibit coherent movements over short distances in time or space. However, in this case, the oil barrels may travel long distances over extended periods, leading to numerous possibilities. Additionally, drifting objects exhibit jibing, the phenomenon of changing direction from parallel to the wind to either leftward or rightward perpendicular and *vice versa* (Breivik et al., 2011). When jibing is accounted for in the simulation, the resulting data comprises stochastic rather than smooth trajectories, making it more challenging to comprehend and analyze. The cylindrical shape of oil barrels makes them particularly susceptible to jibing, making it crucial to implement in the simulation so the most accurate trajectories are obtained.

In order to create more structure within a flock of trajectories and facilitate the analysis of certain patterns, the data needs to be reduced to counteract the effect of the jibing. One method to achieve this is by clustering the trajectories, so significantly similar trajectories are grouped together for the investigation into the offending ship. Clustering can be defined as a method of grouping entities, while maximizing the similarity in a group and minimizing the similarity between groups (Rokach & Maimon, 2005). In

the case of moving objects, similarity can be determined by how closely the objects follow the same trajectories, meaning they are approximately in the same space at each time instant (Rokach & Maimon, 2005). This type of clustering is often referred to as distance-based trajectory clustering, as it compares the distance between trajectories (Giannotti et al., 2008). Another type of clustering is density-based clustering, where a specific number of trajectories within a certain radius are considered similar (Nanni & Pedreschi, 2006). Of each type, two cluster algorithms will be highlighted and explained in detail in sections 1.2.1 and 1.2.2.

1.2.1 Distance-based clustering algorithms

k-Means clustering is a distance-based cluster algorithm that divides a dataset into a predetermined number of clusters, *k*. The algorithm (1) aims to minimize the within-cluster variance by assigning each data point to the cluster with the nearest mean. Each cluster is represented by its centroid, the mean of all data points assigned to that cluster. The algorithm begins by randomly initializing *k* centroids. Each data point is then assigned to the nearest centroid, forming *k* clusters. After all points are assigned, the centroids are recalculated as the mean of the points in each cluster. This process of assignment and updating continues iteratively until the centroids no longer change significantly, indicating convergence (Ahmed et al., 2020).

k-Means is highly efficient, making it suitable for large datasets and capable of handling millions of data points effectively. The linear scalability of *k*-means enhances its practicality for extensive datasets. Additionally, the algorithm converges quickly, often requiring only a few iterations to reach stability. *k*-Means is particularly effective for datasets with clusters that are roughly spherical and evenly sized (Kanagala & Jaya Rama Krishnaiah, 2016). The *k*-means algorithm is visualized in Figure 1.1, where the centroids are displayed as triangles.

Algorithm 1: *k*-Means clustering algorithm

Input : Distance matrix D, number of clusters k
Output: Cluster centroids, cluster assignments

```
Initialize k centroids randomly from D;
while true do
    Assign each data point to the nearest centroid;
    Update each centroid as the mean of the data points assigned to it;
    if no centroid has changed then
        Break;
    end
end
return Cluster centroids, cluster assignments;
```

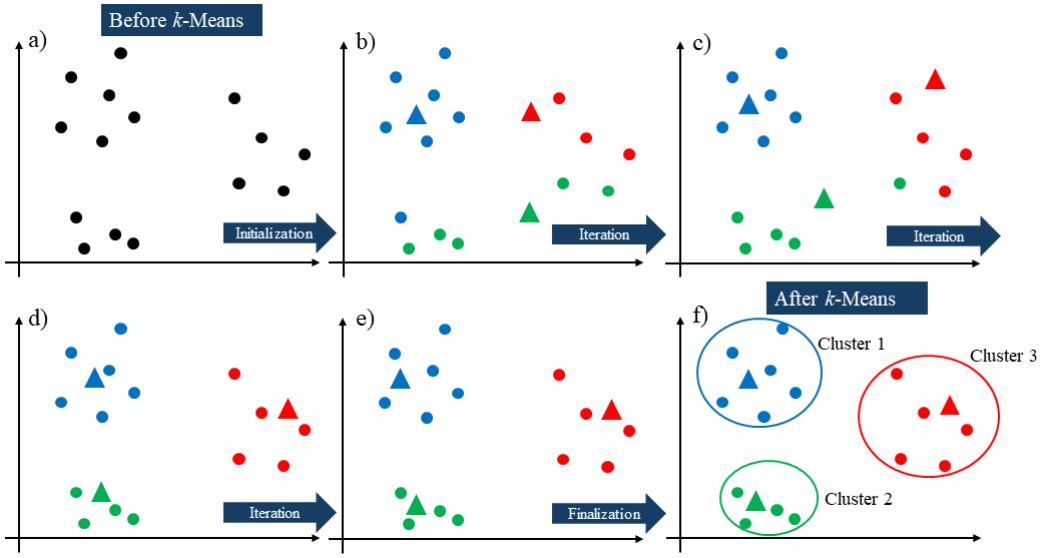


Figure 1.1: Schematic overview of the *k*-means algorithm.

Another distance-based clustering approach is the agglomerative clustering algorithm (2), a hierarchical clustering technique that iteratively merges individual data points or small clusters into larger clusters based on their pairwise similarities. This method starts by treating each data point as a separate cluster and then progressively merges the most similar clusters until all data points belong to a single cluster or until a stopping criterion is met (Murtagh & Contreras, 2011).

Initially, each data point is considered a cluster, forming n clusters, where n is the number of data points in the dataset. The algorithm then calculates the pairwise distances or dissimilarities between clusters using a specified linkage criterion, either single linkage, complete linkage, or average linkage. Single linkage measures the distance between the closest points in each pair of clusters. It focuses on merging clusters that have the closest individual points, often resulting in long, elongated clusters. Complete linkage measures the distance between the farthest points in each pair of clusters. It prioritizes merging clusters whose farthest points are closest to each other, typically producing compact, spherical clusters. Average linkage calculates the distance between the centroids (or means) of the two clusters being merged. It computes the average distance between all pairs of points, one from each cluster. This method provides a balance between single and complete linkage, resulting in clusters that are less elongated than those produced by single linkage but not as compact as those produced by complete linkage (Yim & Ramdeen, 2015; Murtagh & Contreras, 2011).

The merging process continues iteratively by selecting the pair of clusters with the smallest distance according to the chosen linkage criterion and merging them into a single cluster. This step reduces the total number of clusters by one in each iteration, gradually forming a dendrogram or tree structure that illustrates the hierarchical relationships between clusters. The stopping criterion for agglomerative clustering can vary depending on the application. In this research, the algorithm stopped when the predetermined number of clusters k was reached (Murtagh & Contreras, 2011). The agglomerative cluster algorithm is visualized in Figure 1.2.

Algorithm 2: Agglomerative clustering algorithm

Input : Distance matrix D, number of clusters k
Output: Cluster assignments

```
Initialize each data point as a single-point cluster;
Initialize distances between clusters based on chosen linkage criteria;
while number of clusters is not k do
    Merge the two closest clusters;
    Update the distance matrix;
end
return Cluster assignments;
```

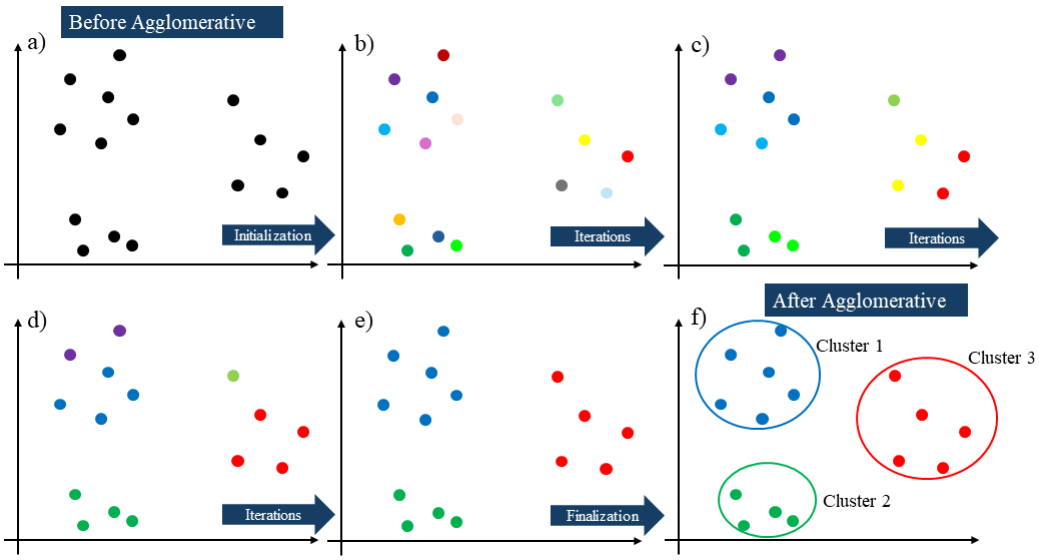


Figure 1.2: Schematic overview of the agglomerative clustering algorithm.

1.2.2 Density-based clustering algorithms

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that builds upon the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) framework by introducing a hierarchical approach (Rahman et al., 2016). This enhancement allows HDBSCAN to identify clusters of varying densities, making it more flexible and robust in comparison to older cluster methods. HDBSCAN relies on the concepts of core distance and mutual reachability distance, as shown in Algorithm 3. The core distance for each point is defined as the distance to its m-th nearest neighbor, where m is a user-defined parameter. This metric serves as an indicator of the local density surrounding an observation. The mutual reachability distance is determined between two points as the maximum of their respective core distances and the direct distance between them. This distance metric effectively smooths the distance landscape by incorporating density information (*Advances in Knowledge Discovery and Data Mining*, 2013; Rahman et al., 2016).

To build the hierarchical structure, HDBSCAN constructs a MST (Minimum Spanning Tree) using the mutual reachability distances as edge weights. This tree captures the hierarchical nature of the data by representing the density-based connectivity between

points. From the MST, a condensed cluster tree is created, which represents clusters at different density levels. This tree encapsulates the hierarchy of clusters, merging clusters as the density threshold decreases, thereby illustrating the clustering structure from the finest to the coarsest scale. The algorithm then extracts a flat clustering from the condensed cluster tree by selecting the most stable clusters. Stability is assessed based on the persistence of clusters across a range of density levels; more stable clusters are considered more significant. This approach allows HDBSCAN to automatically determine the optimal number of clusters without requiring the user to specify this parameter (*Advances in Knowledge Discovery and Data Mining*, 2013; Rahman et al., 2016).

One of the primary advantages of HDBSCAN is its ability to handle clusters with varying densities, which is a limitation in traditional DBSCAN. Additionally, HDBSCAN automatically identifies noise points, enhancing the quality of clustering by excluding outliers (Stewart & Al-Khassaweneh, 2022). The HDBSCAN algorithm is visualized in Figure 1.3.

Algorithm 3: HDBSCAN clustering algorithm

Input : Distance matrix D, minimum number of points m
Output: HDBSCAN hierarchy

Extract the core distance w.r.t. k for all data objects using D;
Compute a MST of G_k, the Mutual Reachability Graph;
Extend the MST to obtain MST_{ext} by adding for each vertex a "self edge" with the core distance of the corresponding object as weight;
Extract the HDBSCAN hierarchy as a dendrogram from MST_{ext};
while MST_{ext} *is not empty* **do**
    Remove all edges from MST_{ext} in decreasing order of weights;
    Set the dendrogram scale value of the current hierarchical level as the weight of the removed edge(s);
    Assign labels to the connected component(s) that contain(s) the end vertex(-ices) of the removed edge(s) to obtain the next hierarchical level;
    Assign a new cluster label to a component if it still has at least one edge, else assign it a null label ("noise");
end

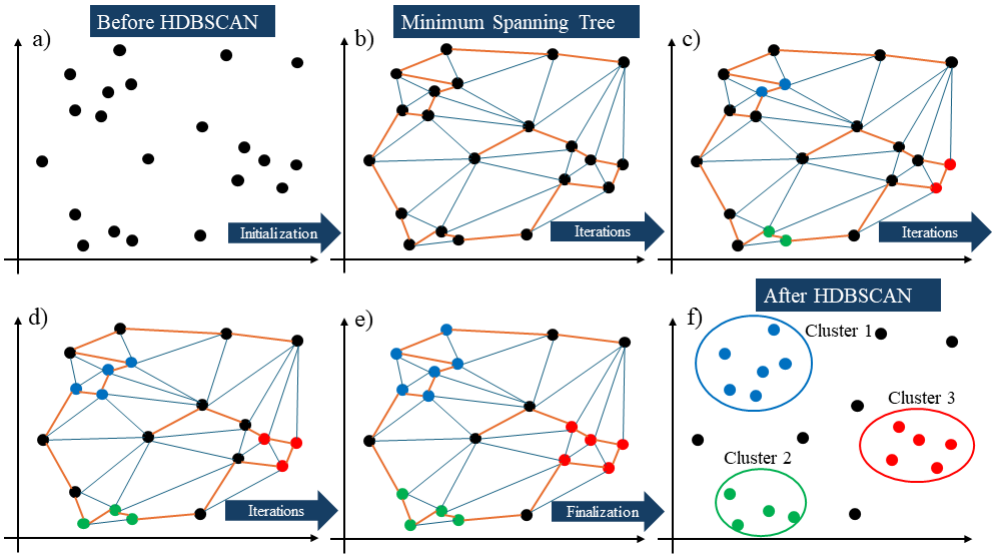


Figure 1.3: Schematic overview of the HDBSCAN algorithm.

Another density-based clustering algorithm is OPTICS (Ordering Points To Identify the Clustering Structure)(4), designed to identify clusters of arbitrary shape and size in large datasets (Ankerst et al., 1999). At the core of OPTICS lies the concept of reachability distance, a metric used to gauge the local density surrounding individual data points. This distance between two points signifies the furthest reach at which one point can be deemed reachable from another, all while adhering to a predefined density threshold. In OPTICS, the parameter ε establishes this maximum distance for reachability, concurrently upholding a specified density threshold. This threshold m denotes the minimal number of points necessary to constitute a cluster. By adjusting ε, the neighborhood size considered during reachability distance computation can be modified, consequently shaping the granularity of the resulting clustering arrangement (Ankerst et al., 1999).

The algorithm calculates the reachability distance for each point with respect to its neighbors. This process generates a reachability plot, which orders the points based on their reachability distances. The reachability plot provides valuable insights into the clustering structure of the dataset, revealing clusters as regions of low reachability distances separated by areas of high reachability distances (Ankerst et al., 1999).

One of the key advantages of OPTICS is its ability to handle datasets with varying densities and noise effectively. By considering the local density of points, OPTICS can adapt to clusters of different shapes and sizes, making it robust to outliers and capable of identifying clusters embedded within clusters (Kanagala & Jaya Rama Krishnaiah, 2016). The OPTICS algorithm is visualized in Figure 1.4, where the core points are displayed as triangles.

Algorithm 4: OPTICS clustering algorithm

Input : Distance matrix D, minimum number of points m, ε-neighborhood radius
Output: Cluster ordering, Reachability plot

```
Initialize core distances to undefined, reachability distances to undefined,
  processed to empty set, and cluster order list to empty;
for each unprocessed point p in X do
    if p is not a core point then
        Mark p as processed;
        Continue to the next point;
    end
    if p is not yet in the cluster order list then
        ExpandClusterOrder(p);
    end
end
Function ExpandClusterOrder(p)
    Add p to the cluster order list;
    for each q in ε-neighborhood of p do
        if q is not processed then
            Calculate r as the maximum of core distance of p and distance between
              p and q;
            if q is not in the cluster order list then
                Add q to the cluster order list;
                if q is a core point then
                    ExpandClusterOrder(q);
                end
            end
        end
    end
return Cluster ordering, Reachability plot;
```

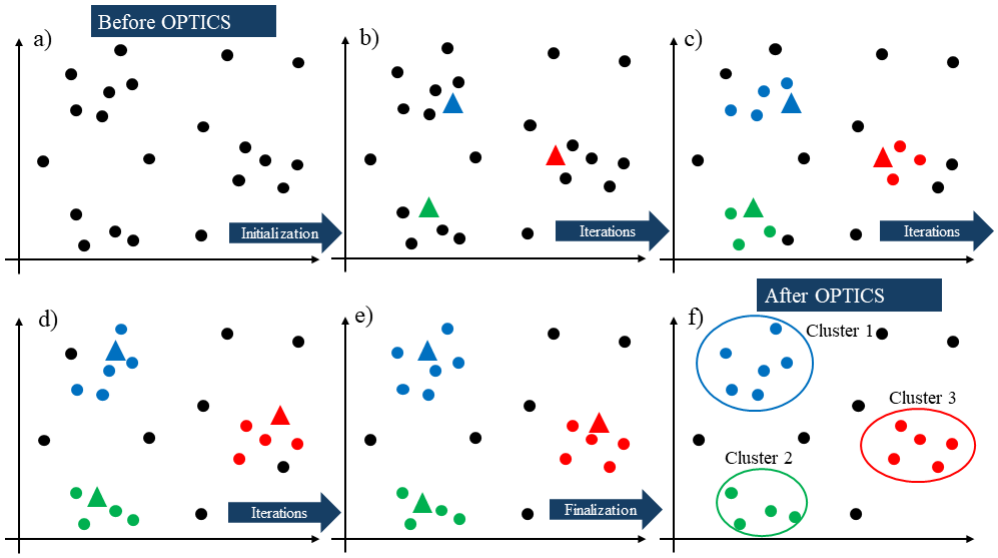


Figure 1.4: Schematic overview of the OPTICS algorithm.

1.3 Research Question

Systematic randomness is introduced into the simulation by the process of jibing, which is modeled by incorporating many ensemble members. This method captures the non-linear effects caused by ocean currents and atmospheric drag. However, this approach also generates a large amount of data that needs to be assessed. Hence, the research question of this project is:

How much data reduction is possible when simulated trajectories are clustered, while maintaining fidelity?

In order to answer this question, the trajectories are simulated forward in time, meaning that in the simulation, the barrels are thrown off a vessel and their trajectories are obtained. Next, the trajectories are simulated backward in time, called an inverse simulation, meaning that from the site location of a barrel, the inverse trajectories are simulated. These two types of simulations are clustered using both distance- and density-based cluster techniques, namely k-means, agglomerative clustering, HDBSCAN and OPTICS, and compared to determine the extent to which the data can be reduced. The data reduction will be quantified in terms of variability.

2. Data

For the simulations, three categories of data were essential. Firstly, data on the site locations of the oil barrels were required. Secondly, data regarding the navigation of ships around these site locations were necessary. Finally, oceanographic models were utilized to simulate the potential trajectories of the oil barrels.

2.1 Oil barrel data

The site locations of the oil barrels were sourced from several entities, including the police department of Terschelling and local residents(Politiebureau Terschelling, 2024). As shown in Figure 2.1, the barrels were sited at locations on Terschelling, Richel, Robbenbank and Ameland.

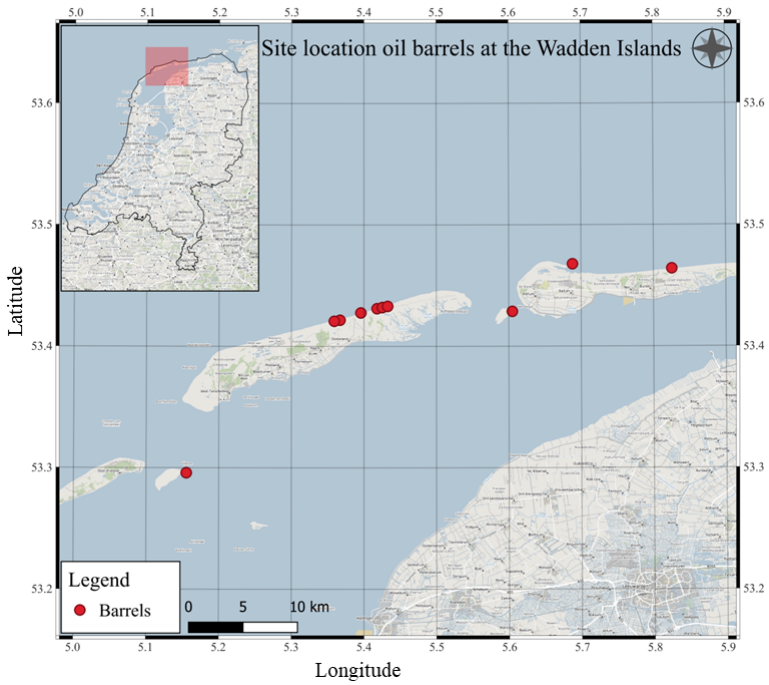


Figure 2.1: Site locations of the oil barrels at the Wadden Islands.

The exact times when the oil barrels drifted ashore are uncertain. Two barrels were found on February 25th at 15:00 and 18:00 UTC (Coordinated Universal Time). The remaining barrels were discovered on February 26th between 10:00 to 14:00 UTC. However, the barrels could have drifted ashore hours or days before being found. The exact site locations are also imprecise. The sources provided vague descriptions, such as *"near beach pole 20"*, which were then converted to coordinates with uncertain accuracy.

2.2 Ship data

The ship data, often referred to as AIS (Automatic Identification System) data, was obtained through Kpler, a company specializing in global trade intelligence (Marine Digital, 2024; Kpler, 2024). The AIS data consists of the latitudes and longitudes of ships along with the corresponding timestamp. The query to Kpler was specified to include only vessels navigating through latitude range 53.4 - 54.7 and longitude range 5.0 - 7.5 between 2023-02-23 00:00 and 2023-02-26 23:59, UTC. This area was chosen based on the northeast wind at the time the barrels were found, suggesting that the origin ship of the oil barrels was navigating northeast of the site locations. The time range was selected due to the uncertainty regarding when the barrels drifted ashore.

The AIS data also includes information such as the MMSI (Maritime Mobile Service Identities) and IMO (International Maritime Organization) numbers, which identify the ship and the owner, respectively (MMSI, 2022; IMO, 2024). In total, the AIS data consisted of 160,794 different coordinates, belonging to 1,029 MMSIs. These ships were associated with 879 unique IMOs and navigated under 51 different flags. Appendix 7.1 includes an overview of each attribute and its corresponding statistics. The coordinates of the vessels at different timestamps were combined to determine their trajectories, as shown in Figure 2.2.

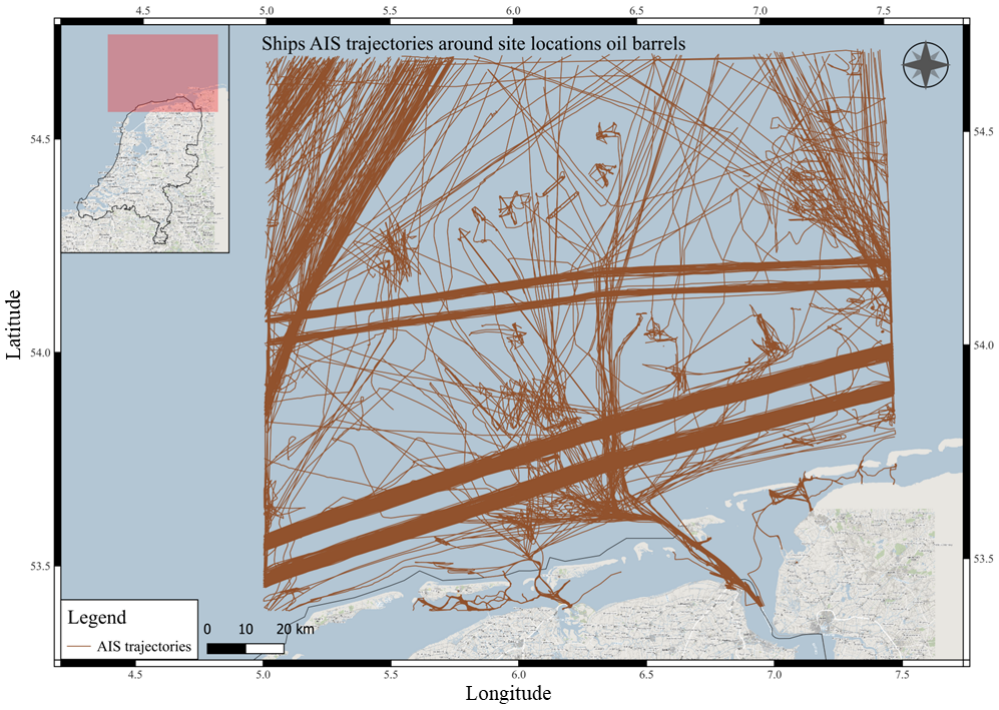


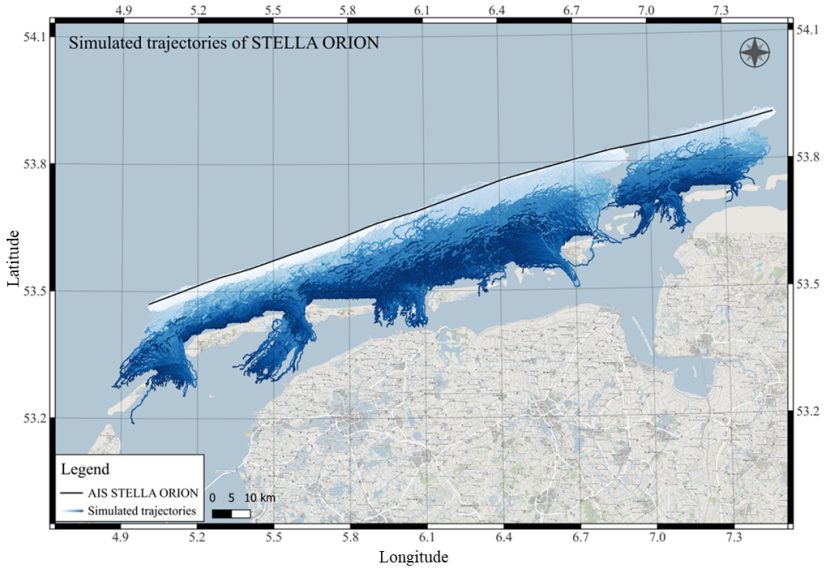
Figure 2.2: AIS trajectories of ships navigating through latitude range 53.4 - 54.7 and longitude range 5.0 - 7.5 between 2023-02-23 00:00 and 2023-02-26 23:59, UTC.

2.3 Oceanographic data

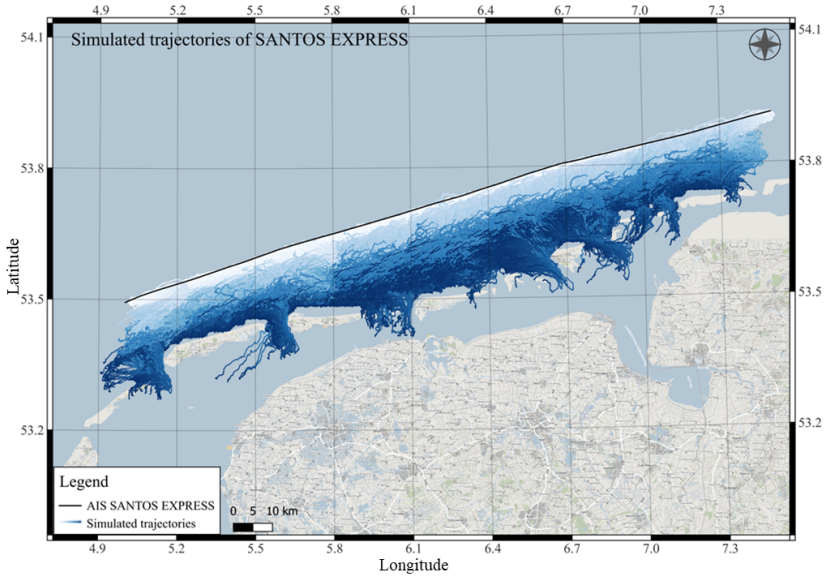
The simulations were conducted using oceanographic models, specifically an atmospheric model, a wave model and an ocean model. The wave model, SWAN (Simulating WAves Nearshore), was developed at Delft University of Technology and provides estimations of wave conditions during specific times ("The SWAN team: SWAN - Scientific and technical documentation SWAN Cycle III version 41.20A", n.d.). This model covers the North Sea and part of the northeast Atlantic, with a horizontal resolution of approximately 5 kilometers (Sterl & Ministry of Infrastructure and Water Management, 2019). The atmospheric model, Harmonie, originates from the KNMI (Royal Netherlands Meteorological Institute) (KNMI, 2024; Sterl & Ministry of Infrastructure and Water Management, 2019). Harmonie covers the same domain as SWAN with a higher horizontal resolution of 2.5 kilometers (Sterl & Ministry of Infrastructure and Water Management, 2019). For the ocean model, the 3D DCSM-FM (three-dimensional Dutch Continental Shelf Model - Flexible Mesh), version 6, was used, designed by RWS (Rijkswaterstaat) and Deltares (RWS & Deltares, 2022; RWS, 2024; Deltares, 2024).

2.4 Cluster data

The research conducted by Van der Minnen (2024) focused on identifying the vessel responsible for littering the oil barrels by simulating their trajectories and intersecting them with AIS trajectories (Van der Minnen, 2024). From Van der Minnen's list of the ten most likely suspects, two vessels were chosen for the forward simulation. Figure 2.3a shows the simulated trajectories of the oil barrels if they were littered from the vessel with MMSI 246553000, an oil and chemical tanker called *STELLA ORION*, navigating under the Dutch flag (kpler, n.d.-b). The forward simulation using *STELLA ORION* consists of 4000 trajectories each with 500 timestamps. Due to hardware computation limitations, a selection was made of 667 trajectories by including every sixth trajectory. Figure 2.3b shows the simulated trajectories of barrels if they were littered from the vessel with MMSI 218854000, a container ship called *SANTOS EXPRESS*, navigating under the German flag (kpler, n.d.-a). The forward simulation using *SANTOS EXPRESS* consists of 3720 trajectories each with 500 timestamps. A selection of 620 trajectories was made by including every sixth trajectory. Both simulations had a maximum time span of 38 hours, with a timestamp interval of 5 minutes for each trajectory.



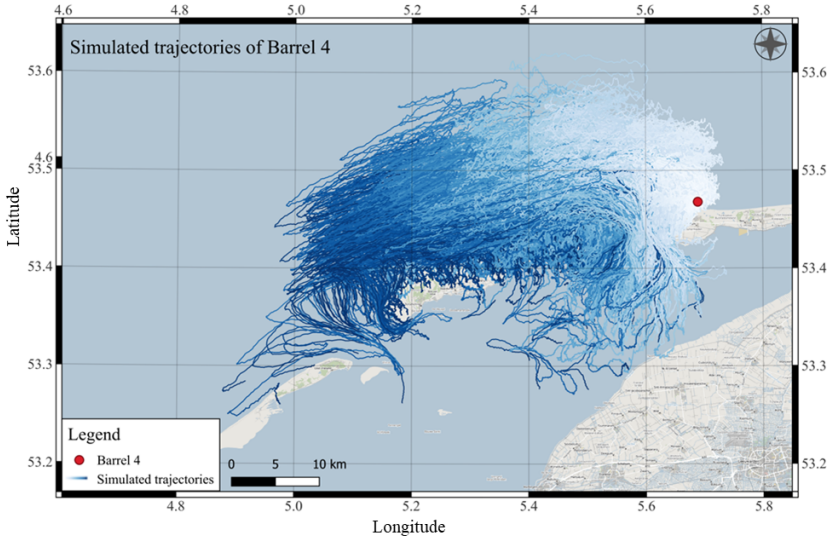
(a) *STELLA ORION*



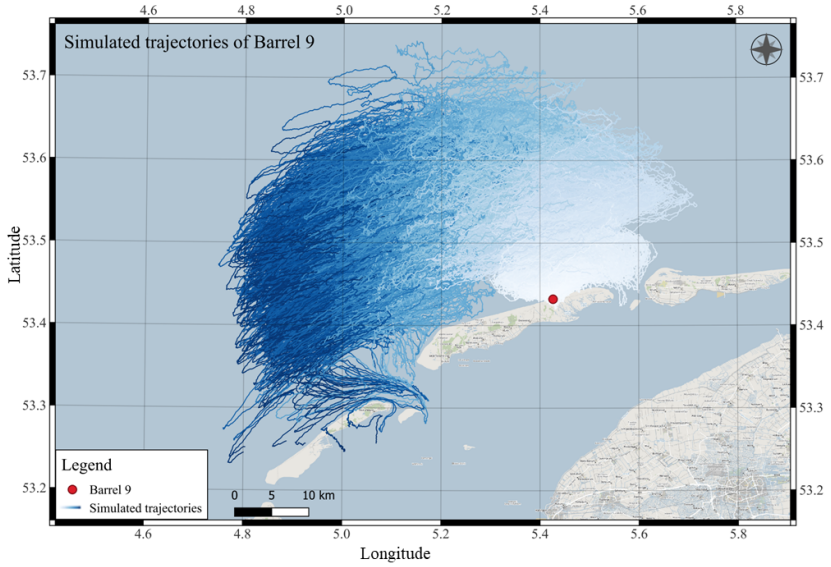
(b) *SANTOS EXPRESS*

Figure 2.3: The forward simulated trajectories using vessels *STELLA ORION* and *SANTOS EXPRESS* with their AIS represented as a black line. The colors represent the progression of time, where light indicates the start of the trajectory and dark indicates the end.

The inverse simulation of the trajectories of two barrels, identified as 4 and 9, were used for clustering. Barrel 4 was found at Ameland, north of Balm, on February 25th around 15.00 UTC. Barrel 9 was sited at Terschelling *"around beachpole 20.7"* on February 26th. These barrels were chosen because they were found on different days and islands, maximizing the spatio-temporal spread. Barrel 4's and 9's inverse simulated trajectories are shown in Figure 2.4a and 2.4b, respectively. The inverse simulation using Barrel 4 consists of 500 trajectories, each with 1050 timestamps and a maximum time span of 85 hours and the simulation using Barrel 9 consists of 500 trajectories, each with 1300 timestamps and a maximum time span of 106 hours. Both simulations were run with a timestamp interval of 5 minutes for each trajectory. The two simulations did not need a pre-selection since they contained a relatively small number of trajectories.



(a) Barrel 4



(b) Barrel 9

Figure 2.4: The inverse simulated trajectories of Barrel 4 and 9, with the barrels represented as dots. The colors represent the progression of time, where light indicates the start of the trajectory and dark indicates the end.

As an overview, the two AIS trajectories for the forward simulation and the two barrels for the inverse simulation are displayed together in Appendix B in Figure 7.1. The input parameters for the forward and inverse simulations are detailed in Appendix B as Table 7.2. The standard deviation and mean absolute difference (MAD) of all four simulations are provided in Appendix B as Table 7.3 (Song et al., 2003). The simulations of the remaining eight AIS trajectories of Van der Minnen's list and the remaining nine barrels were clustered as a reference.

3. Methods

3.1 Pre-processing

The computation of cluster centers and prediction of cluster indexes for each trajectory required a distance matrix as input. This distance matrix was obtained by first transforming the trajectories into a three-dimensional array with the following dimensions:

(N^o trajectories, N^o observations, 2)

This array represents the coordinates of each observation of a trajectory. Due to the nature of the simulations, each trajectory had the same amount of observations, each made with the same time interval. The distance matrix was obtained by calculating the average Euclidean distance between the trajectories, as defined with Formula 3.1 (Krislock & Wolkowicz, 2012). Between two trajectories, the distance was determined by averaging the Euclidean distance between coordinate pairs. These pairs share the same index within the trajectory; hence, the distance was calculated only between the first and first coordinate observation, second and second, and so on. In this way, the temporal component was taken into account, making the distance measure based on how well the trajectories follow the same path over time. Any missing data was disregarded in this calculation, so if a trajectory started later or ended earlier by stranding on a shore, only the overlapping part of the two trajectories in time was considered.

$$D(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{3.1}$$

where $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$ are the coordinates of the observations in the n-dimensional space. This method was performed for each simulation, both forward and inverse.

3.2 Input Parameters

Four cluster algorithms were implemented on each simulation: two distance-based clustering approaches, k-means and agglomerative clustering and two density-based clustering approaches, HDBSCAN and OPTICS. The cluster algorithms were all included in the *Scikit-learn* Python package (Pedregosa et al., 2011).

The k-means input parameter for the number of clusters k, was varied between 2 and 10, as recommended by Ahmed et al. (2020). The complete, average and single linkage were implemented for the agglomerative clustering approach. For all three linkage methods, the number of clusters k was varied between 2 and 10, as recommended by Yim and Ramdeen (2015). For HDBSCAN, the input parameter for the minimum number of observations in a cluster, m, was varied between 5 and 75, in increments

of 5, as recommended by Rahman et al. (2016). The minimal number of points in a cluster m for OPTICS was again varied between 5 and 75, in increments of 5, as recommended by Ankerst et al. (1999). The neighborhood radius ε ranged between 0 and 1, in increments of 0.1. Of each clustering approach, the silhouette score was calculated for every parameter setting with Formula 3.2.

Of each model, the silhouette score (S) was calculated according to Formula 3.2 (Rousseeuw, 1987).

$$S = \frac{1}{N} \sum_{i=1}^{N} \frac{D_i^{\min} - \bar{D}_i}{\max\{\bar{D}_i, D_i^{\min}\}} \tag{3.2}$$

where N is the number of observations, $\bar{D}_i$ is the average Euclidean distance (3.1) from the ith observation to other observations in the same cluster, and $D_i^{\min}$ is the smallest average distance from the ith point to points in a different cluster. The silhouette score can range between -1 and 1, where a negative value indicates that the clusters have a greater dissimilarity within a cluster than between clusters (Rousseeuw, 1987). A high positive value for S, indicates that the observations have a great chance to be clustered correctly (Shutaywi & Kachouie, 2021) (Rousseeuw, 1987). There is a consensus that a clustering approach should have a silhouette score higher than 0.5 to be considered as well-performing (Rousseeuw, 1987).

3.3 Comparison

For each clustering approach, the parameter settings that yielded the highest silhouette score, and thus the best performance, were selected for further implementation (Rousseeuw, 1987). The clusters were then presented along with their medoids, facilitating an understanding of the central tendency of trajectories within clusters, which is essential for gaining insights into the underlying structure of the trajectory data (Estivill-Castrol & Murray, 1998). The medoid of a cluster (M) was defined as shown in Formula 3.3 (Jimoh et al., 2022).

$$M = \min \sum_{j=1}^{T} D(M, t_j) \tag{3.3}$$

where T is the number of trajectories in the cluster and D the Euclidean distance (3.1) between M and t_j, the jth trajectory of the cluster.

The data reduction was quantified by comparing the variability of the data before and after clustering. The general equation used is Formula 3.4.

$$R = (1 - \frac{var_{\text{before}}}{var_{\text{after}}}) * 100\% \tag{3.4}$$

where R is the reduction in percentage and $\frac{var_{\text{before}}}{var_{\text{after}}}$ the ratio of a variability measure before and after clustering. The standard deviation and the MAD were used as measures. The standard deviation was calculated by first determining the average standard deviation within a cluster and then averaging these values (Formula 6.3). This was also done using the weighted average standard deviation (Formula 6.4) where the average standard deviation within a cluster was weighted based on the number of trajectories in the cluster.

Since the standard deviation squares the differences between points and the mean, giving more weight to outliers and noise, the average MAD was used as a second, more robust measure of variability (Formula 6.6) (Pastor & Socheleau, 2012). This process was repeated using the weighted average MAD, where the average MAD within a cluster was weighted with the number of trajectories in the cluster (Formula 6.7). These four approaches lead to the following measures of data reduction:

1. R_V using the average standard deviation (Formula 6.8)
2. R_V^{weighted} using the weighted average standard deviation (Formula 6.9)
3. R_{MAD} using the average MAD (Formula 6.10)
4. $R_{\text{MAD}}^{\text{weighted}}$ using the weighted average MAD (Formula 6.11)

These measures indicate how well the variability was reduced using clustering techniques. A high reduction metric suggests homogeneity within the clusters and distinctiveness between the clusters. Therefore, a high reduction value indicates that the medoids are effective for representing the clusters and reducing the data. It is expected that the clustering methods that perform the best will also show the most reduction.

4. Results and Analysis

4.1 Cluster Performance

In Appendix B, the change in silhouette scores during the parameter tuning is shown for all simulations. For the *STELLA ORION* simulation (Figure 7.2), *k*-Means reached its peak silhouette score with 3 and 4 clusters, after which the silhouette score decreased with increasing *k*. The average and complete linkage methods for agglomerative clustering showed relatively consistent silhouette scores around 0.50. The single linkage method had a high silhouette score for 2 clusters, but the scores became negative as the number of clusters increased. The highest silhouette score for the complete linkage method was 0.610 with 3 clusters. HDBSCAN displayed consistent silhouette scores for minimum cluster sizes ranging from 0 to 60. OPTICS exhibited no variation in silhouette scores for different values of ε, achieving the highest silhouette scores for minimum cluster sizes of 40, 50, 55, 65, and 70. The highest silhouette score for HDBSCAN and OPTICS were both 0.532. Since all silhouette scores were above the threshold of 0.50, these four clustering approaches can be considered to perform well based solely on the silhouette score (Shutaywi & Kachouie, 2021).

For the *SANTOS EXPRESS* simulation (Figure 7.3), *k*-Means showed a smooth decrease in silhouette scores to 0.512 as *k* increased, with the the highest silhouette score of 0.628 achieved at the minimal value of *k*, 2. As with *STELLA ORION*, the single linkage method for the agglomerative yielded negative silhouette scores while complete and average had relatively consistent scores around 0.50. The highest silhouette score was 0.628, obtained with the average linkage method and 2 clusters. HDBSCAN resulted in only one cluster for several values of the minimum cluster size. As the silhouette score requires at least two clusters, complete data on silhouette score changes during parameter tuning could not be obtained. Only parameters resulting in more than one cluster were considered for this research, since it is focused on data reduction through clustering. The highest silhouette score for HDBSCAN fulfilling this requirement was 0.318. OPTICS again showed no variation in silhouette scores for different values of ε, with positive silhouette scores for minimum cluster sizes of 25 and 40, the latter resulting in the highest score of 0.437. For the *SANTOS EXPRESS*, only *k*-means and agglomerative clustering exceeded the threshold silhouette score of 0.50, indicating good performance for these methods, whereas OPTICS and HDBSCAN did not perform as well (Shutaywi & Kachouie, 2021).

The inverse simulations of Barrel 4 and 9 (Figure 7.4 and 7.5) encountered similar issues with HDBSCAN as with *SANTOS EXPRESS*, with several parameter settings resulting in only one cluster. These outcomes were disregarded. The highest silhouette scores with HDBSCAN for Barrel 4 and 9 were 0.09 and 0.052, respectively. Variation in ε for OPTICS did not result in different silhouette scores for either barrel. Barrel 4 had the highest, although still negative, score of -0.353 with a minimum cluster size of 5.

Higher parameter settings resulted in a silhouette score of -1.00. Barrel 9 followed a similar pattern, with the highest silhouette score of -0.226 at a minimum cluster size of 15. Both barrels showed similar trends in silhouette scores for agglomerative clustering and *k*-means. With 2 clusters and average linkage approach, agglomerative clustering achieved the highest silhouette scores of 0.428 and 0.421 for Barrels 4 and 9, respectively. For, *k*-means, the silhouette scores peaked at the minimal value of *k*, 2, resulting in scores of 0.436 and 0.438 for Barrels 4 and 9, respectively. None of the clustering methods for the inverse simulations achieved a silhouette score exceeding the threshold of 0.50, indicating poor performance across all methods (Shutaywi & Kachouie, 2021).

There is a clear distinction visible between the distance-based and density-based approaches. For all simulations, the distance-based approaches performed better, even though they are theoretically less robust for noise and outliers (Kanagala & Jaya Rama Krishnaiah, 2016). The forward simulations were, in general, better suited for clustering than the inverse simulations. This can be attributed to the data structure of the trajectories. The flocks of the forward simulations were initially more stretched out in the longitude direction, as can be seen in their initial standard deviation, Appendix 7.3. The inverse simulated flocks had almost double the standard deviation in the latitude direction but a smaller one in the longitude direction, Appendix 7.3. This difference leads to a more chaotic flock for the inverse simulations, making them harder to cluster (Nanni & Pedreschi, 2006).

4.2 Data Reduction

In Figure 4.1, the clusters and their medoids are plotted for the forward *STELLA ORION* simulation. HDBSCAN and OPTICS yielded identical cluster results. The distinct group of trajectories on the right was clustered together, containing 113 trajectories. The remaining 554 trajectories were grouped into a single cluster. Agglomerative clustering and *k*-means also clustered the right group of 113 trajectories together. However, they differed in how they clustered the remaining trajectories. Agglomerative clustering divided the remaining trajectories into two clusters of 298 and 256 trajectories, respectively, from left to right. In contrast, *k*-Means divided the remaining trajectories into three clusters of 194, 185 and 175 trajectories, from left to right. Since *k*-means had the highest performance based on the silhouette scores, the medoids of these clusters are considered the most effective for data reduction.

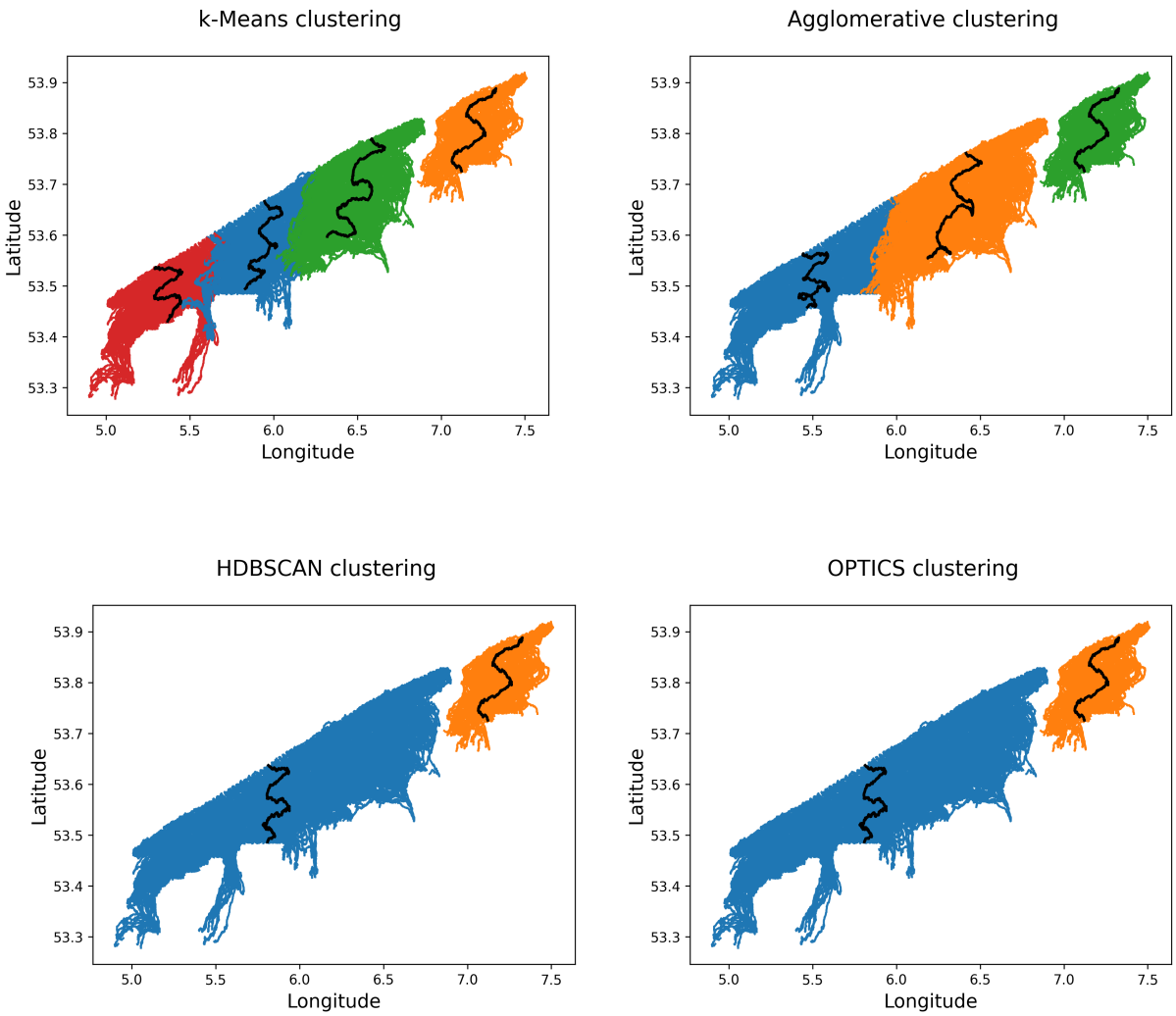


Figure 4.1: The clustering results of *STELLA ORION*, with each cluster plotted in a different color. The medoid of each cluster is visualized as a black line.

In Figure 4.2, the clusters and their medoids are depicted for the forward *SANTOS EXPRESS* simulation. The four clustering methods divided the trajectories into two groups. HDBSCAN and OPTICS have considered 147 and 84 trajectories as noise, respectively, which are represented in Figure 4.2 as black clusters. HDBSCAN's clusters contained 99 and 374 trajectories, from left to right, while OPTICS' clusters contained 163 and 373 trajectories. Agglomerative clustering and k-means produced similar results, with k-means clusters having sizes of 286 and 334, and agglomerative clusters having sizes of 292 and 328. Since both methods achieved the highest silhouette scores, the medoids from either method can be considered effective for data reduction.

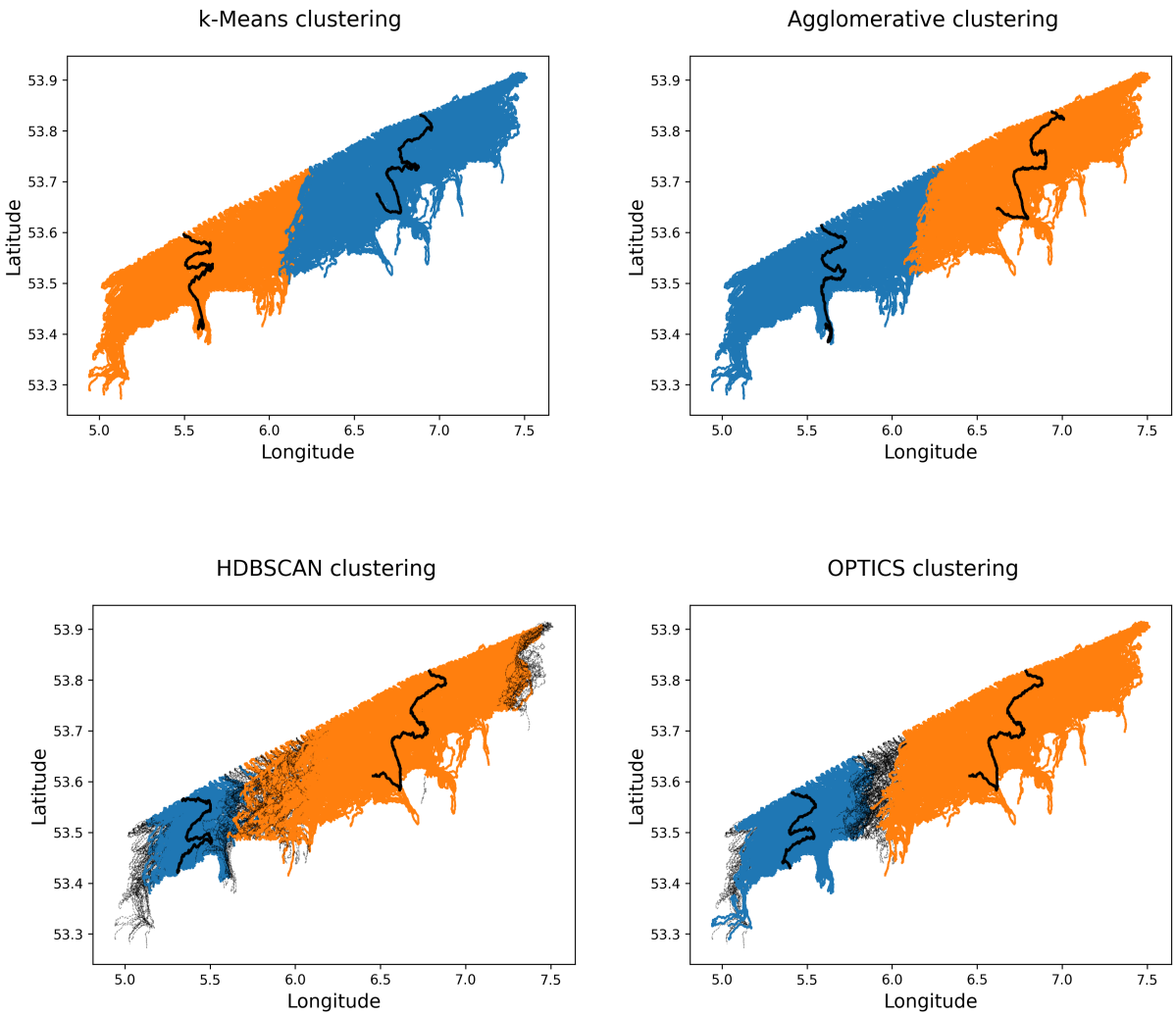


Figure 4.2: The clustering results of *SANTOS EXPRESS*, with each cluster plotted in a different color. The medoid of each cluster is visualized as a black line and the noise trajectories as thin, dashed black lines.

In Figure 4.3, the clusters for the inverse Barrel 4 simulation are shown, with their medoids highlighted in black. Due to overlapping and noise, these plots are relatively difficult to interpret, so the clusters are also plotted separately in Appendix 7.4. *k*-Means and agglomerative clustering appeared to have the same medoids although their clusters are not exactlty the same. Agglomerative clustering resulted in clusters of 280 and 220 trajectories from left to right, whereas *k*-means resulted in clusters of 290 and 210 trajectories. HDBSCAN displayed similar cluster patterns with 142 and 151 trajectories, while 202 trajectories were considered noise. The OPTICS algorithm resulted in 11 clusters, each ranging in size from 5 to 11 trajectories. More than 80% of the data was considered noise, amounting to 407 trajectories.

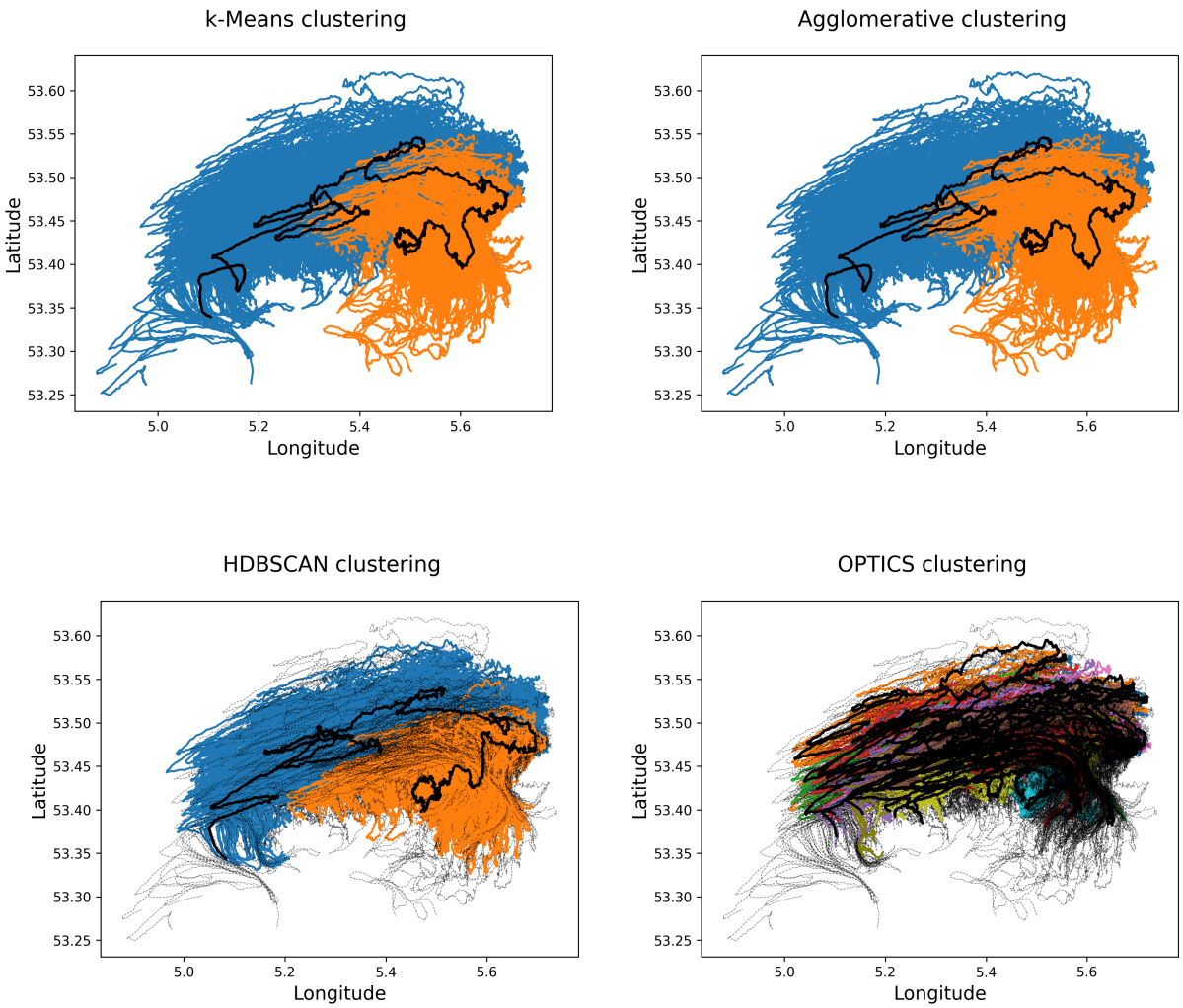


Figure 4.3: The clustering results of Barrel 4, with each cluster plotted in a different color. The medoid of each cluster is visualized as a black line and the noise trajectories as thin, dashed black lines.

Figure 4.4 shows the clusters of the inverse Barrel 9 simulation, with the separate clusters plotted in Appendix 7.4. *k*-Means and agglomerative clustering again displayed the same cluster medoids, with sizes of 364 and 136 for *k*-means and 351 and 149 for agglomerative clustering. HDBSCAN identified relatively similar clusters with sizes of 305 and 50 trajectories, a third small cluster containing 5 trajectories and 140 noise trajectories. OPTICS, similar to its performance with Barrel 4, identified a high percentage of noise, amounting to 450 trajectories, and three small clusters, each containing around 16 trajectories.

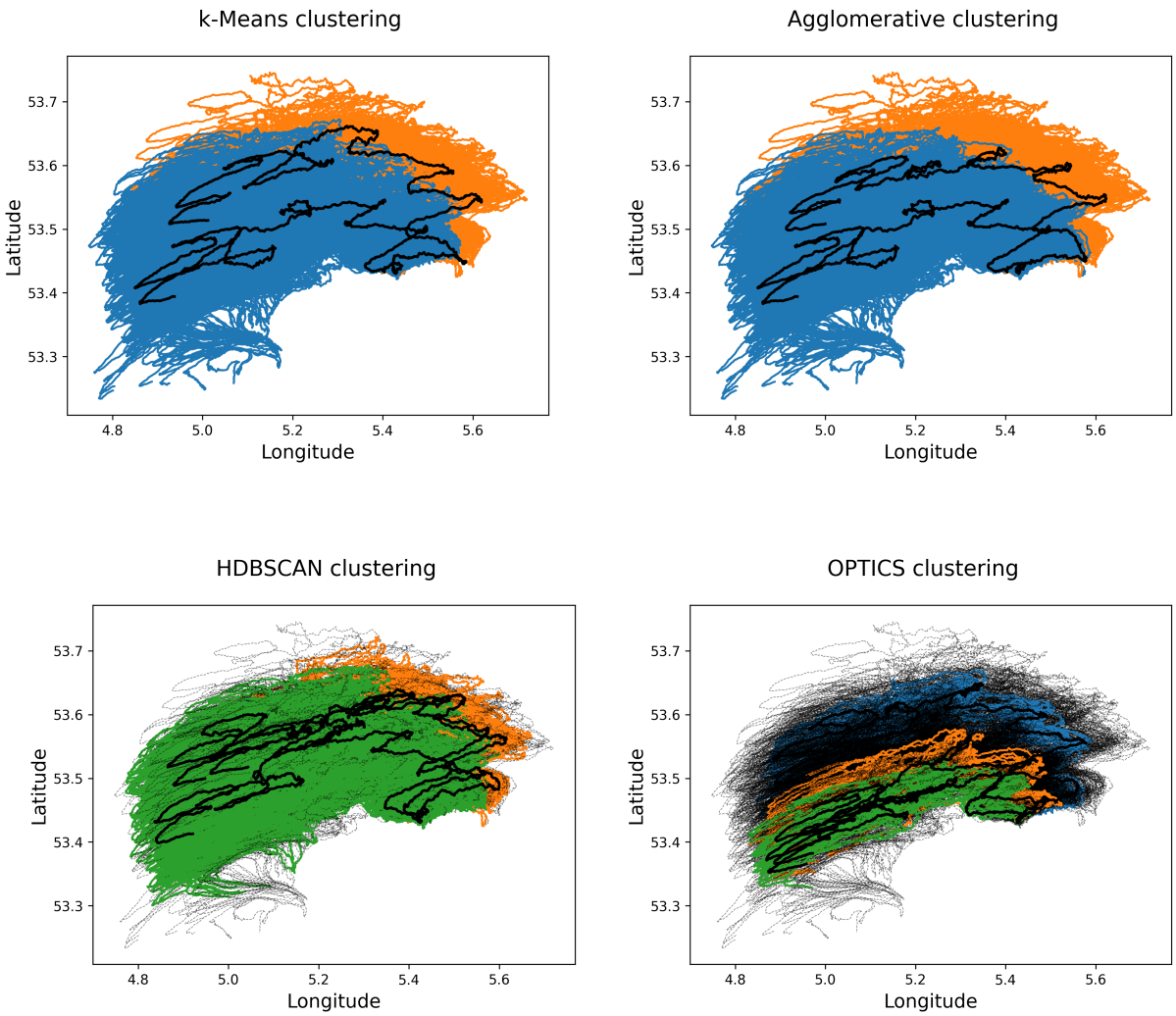


Figure 4.4: The clustering results of Barrel 9, with each cluster plotted in a different color. The medoid of each cluster is visualized as a black line and the noise trajectories as thin, dashed black lines.

In Table 4.1, the reduction in variability based on the average and weighted average standard deviations are shown. For the forward *STELLA ORION* dataset, clustering methods demonstrated limited effectiveness, yielding reductions in variability ranging from near zero to very small negative values (approximately -0.003% to -0.72%). The *SANTOS EXPRESS* simulation showed a negative reduction, indicating an increase in variability for the density-based approaches. However, when considering the weighted reduction in variability, these methods showed relatively good results, with reductions around 20% and 12%. This difference is due to the noise trajectories, which decrease the standard deviation within the clusters, thereby reducing the overall variability. For the forward *SANTOS EXPRESS* simulation and the inverse simulations for Barrel 4 and 9, the distance-based approaches again showed reductions near zero for both R_V and R_V^{weighted}. In contrast, the density based approaches showed high reductions for the inverse simulations. OPTICS, in particular, showed values around 80% and 90% for R_V^{weighted}. These values are to be expected, since OPTICS filtered more than 80% of the trajectories by considering them as noise. In general, the variability was more effectively reduced in the latitude direction then in the longitude.

Table 4.1: Reduction of variability based on the average and weighted average standard deviations.

Simulation	Cluster method	R_V (%)		R_V^{weighted}(%)	
		Longitude	**Latitude**	**Longitude**	**Latitude**
Forward *STELLA ORION*	*k*-Means	0.000	-1.192 E-05	3.838 E-06	-1.610 E-05
	Agglomerative	1.192 E-06	-1.192 E-05	4.112 E-06	-2.347 E-05
	HDBSCAN	1.192 E-05	-1.192 E-05	1.156 E-05	-1.566 E-05
	OPTICS	1.192 E-05	-1.192 E-05	1.156 E-05	-1.566 E-05
Forward *SANTOS EXPRESS*	*k*-Means	2.384 E-05	1.192 E-05	2.298 E-05	9.604 E-06
	Agglomerative	3.576 E-05	1.192 E-05	3.034 E-05	1.663 E-05
	HDBSCAN	-4.464	-0.651	20.30	23.21
	OPTICS	-2.635	-0.030	11.27	13.52
Inverse Barrel 4	*k*-Means	0.000	7.153 E-05	-5.304 E-06	6.795 E-05
	Agglomerative	0.000	7.153 E-05	-9.430 E-07	7.332 E-05
	HDBSCAN	0.1778	4.468	41.50	44.02
	OPTICS	-16.42	4.823	78.35	82.30
Inverse Barrel 9	*k*-Means	-7.153 E-05	-3.576 E-05	-6.658 E-05	-3.284 E-0
	Agglomerative	-7.153 E-05	-3.576 E-05	-6.122 E-05	-3.539 E-05
	HDBSCAN	0.332	4.523	28.24	31.27
	OPTICS	4.008	7.901	90.40	90.79

Table 4.2 shows the reduction of variability based on the MAD and weighted MAD. For the forward *STELLA ORION* simulation, the weighted averages indicated a slight improvement in clustering effectiveness when considering the entire dataset, with weighted average reductions in variability ranging from approximately 0.089% to 0.298%. The *SANTOS EXPRESS* simulation showed negligible reduction for the distance-based approaches and relatively good results for the density-based approaches, considering the $R_{\text{MAD}}^{\text{weighted}}$. The same patterns were visible for the two inverse simulations. HDBSCAN showed the best results for *SANTOS EXPRESS*, whereas OPTICS displayed the highest reduction for the inverse simulations. These results are expected since HDBCAN had the largest noise cluster for *SANTOS EXPRESS* and OPTICS for the inverse simulations. The variability was again more reduced in the latitude direction then in the longitude, although the differences were small.

Table 4.2: Reduction of variability based on the MAD and weighted MAD.

Simulation	**Cluster method**	$R_{\mathbf{MAD}}$ (%)		$R_{\mathbf{MAD}}^{\mathbf{weighted}}$ (%)	
		Longitude	**Latitude**	**Longitude**	**Latitude**
Forward *STELLA ORION*	*k*-Means	0.180	0.424	0.089	0.298
	Agglomerative	0.428	0.568	0.218	0.229
	HDBSCAN	0.715	0.986	0.021	0.290
	OPTICS	0.715	0.986	0.021	0.290
Forward *SANTOS EXPRESS*	*k*-Means	0.179	0.038	0.002	0.0003
	Agglomerative	0.132	0.025	-0.003	-0.001
	HDBSCAN	-3.508	0.101	21.98	23.57
	OPTICS	-0.931	0.232	12.62	13.53
Inverse Barrel 4	*k*-Means	3.823	0.387	1.008	0.102
	Agglomerative	2.872	0.314	0.940	0.103
	HDBSCAN	2.008	0.527	43.06	41.81
	OPTICS	-13.43	-1.009	79.58	81.52
Inverse Barrel 9	*k*-Means	0.011	-0.059	-0.001	0.006
	Agglomerative	0.0422	-0.072	-0.006	0.010
	HDBSCAN	0.595	-0.603	27.94	28.08
	OPTICS	-0.638	-0.519	89.93	89.95

5. Discussion and Conclusion

5.1 Discussion

The answer to the research question, to what extent the data can be reduced through clustering, depends on whether or not you want to include every trajectory of the data. The general observation is that the inverse simulations showed the highest reduction in variability. The density-based approaches showed significantly better results than the distance-based methods. However, this is a direct result of their ability to categorize trajectories as noise (Kanagala & Jaya Rama Krishnaiah, 2016). This is not favored for every implementation. In the case of identifying the offending ship, excluding more than 80% of the possible trajectories could lead to very different results and, in worst case, wrong accusations. Therefore, if it is crucial to include all the trajectories, OPTICS and HDBSCAN should not be used. In that case, the answer to the research question is that the variability of the data in terms of standard deviation could not be reduced and in terms of the MAD only with a maximum of 0.229 %. It can be questioned if this number is worth the trouble of clustering the data. If it is not crucial to include every trajectory or maybe even favored, OPTICS and HDBSCAN should be used. Then, the data can be reduced significantly with the best results obtained with the OPTICS algorithm.

In this research, the cluster performance is only expressed in terms of the silhouette score. Since the score uses the distance between observations in the cluster, the score always favors cluster approaches that minimizes the distance between points in a cluster (Rousseeuw, 1987). This was also observed in the results; Purely based on the silhouette score, the distance-based approaches always performed better than the density-based methods. In order to measure the performance of a cluster method unbiased, it is preferable to use several performance measures. For spatio-temporal data, the performance metrics that can be used for unsupervised learning are all using distance values. Other performance metrics such as CHI (Calinski-Harabasz Index) or DBI (Davies-Bouldin Index) are not applicable to spatio-temporal data (Wang & Xu, 2019; Petrovic, 2006). Therefore, a ground truth of the clusters should be implemented to enable the use of measures such as the mean squared error or Adjusted Rand Index (Santos & Embrechts, 2009; Rezaie & Saunier, 2021). Due to time limitations, this was not done during this research. However, it is recommended for future studies to identify a certain ground truth of clusters. This can be done, for instance, by clustering the trajectories based on the starting time of the trajectory for the forward simulations, or the length of the trajectory for the inverse simulations (Rezaie & Saunier, 2021).

A second effect of the limited time frame of this research was that only four clustering mechanisms have been implemented. These were deliberately chosen, based on literature research. However, there are more methods that could show interesting results. For instance, a third clustering technique, model-based, could show different

results. Model-based clustering aims to cluster data based on their probability distribution (Bouveyron & Brunet-Saumard, 2014). Although this method is mostly used on one-dimensional data, studies are expanding the use of existing software to high-dimensional data, making it an interesting option to include in future studies (Bouveyron & Brunet-Saumard, 2014).

As a third limitation due to time shortage, the data reduction was only expressed in two variability measures, standard deviation and MAD. Based on these two metrics, a conclusion was drawn on whether or not clustering was effective for data reduction. Implementing more measures, such as the interquartile range, could offer different perspectives on the reduction of variability of the data.

There are three additional methods that should be considered in future research regarding the distance matrix. Firstly, the distance measure used does not scale the longitude and latitude according to the curvature of the Earth. Given that the simulations cover a relatively small portion of the Earth's surface and are situated far from the poles, it is unlikely that this omission would significantly impact the results. The effect of the Earth's curvature is anisotropic, causing distortion primarily in the east-west direction rather than the north-south direction. Therefore, it is important to implement longitude scaling for more extensive simulations or for those conducted at different latitudes. This would ensure accuracy in distance measurements, particularly for larger geographic areas or locations closer to the poles. Next, the distance measure is established by defining the Euclidean distance between coordinate pairs within a trajectory (see Chapter 3). This method bases the distance between trajectories on the moment of barrel release for the forward simulations. By subtracting the mean position of the trajectories from the coordinates, trajectories following the same path will yield a smaller distance measure, indicating greater similarity. This approach should be implemented in future research, as it will likely result in different clusters, which could be useful for further data reduction. Lastly, only the Euclidean distance metric was used in this study. Incorporating other metrics, such as the Hausdorff or Fréchet distance, could provide additional insights into cluster patterns and would be a valuable addition to future analyses (Wai & Nwe, 2017).

If all these enhancements are fulfilled, it is unlikely that the answer to the research question will be significantly different. Additional performance measures could lead to a more reliable insight into whether the medoid can be used as a form of data reduction. Within the scope of this research, representing the data of the forward simulation of *STELLA ORION* as two medoids would probably not be useful for investigating whether the eleven oil barrels originated from *STELLA ORION*. Even if another clustering approach and more variability measures are implemented, the clusters are only useful if they can help in identifying the vessel responsible for the pollution. The number of clusters for the distance-based approaches can be specified and in this research, they showed acceptable results for ten clusters, Appendix 7.3b. Subsequently, the medoids of the ten clusters could be used as a form of data reduction. As stated in Chapter 3, it was expected that if a method had a higher silhouette score, it would reduce the variability more. This was proven for the distance-based approaches, but not for the density-based models due to the bias in the silhouette score. The distance-based methods with 10 clusters performed worse than the models discussed in Chap-

ter 4. Hence, the reduction in variability is expected to be lower, even closer to zero. Therefore, it is debatable whether data reduction through clustering will yield useful results for this specific type of application.

Future research should incorporate AIS trajectories of vessels located further from the shoreline and perform simulations for the inverse models using longer trajectories. These modifications will result in larger and more chaotic flocks. Conversely, shortening the time span should also be explored, as it will produce smaller and less chaotic flocks. The density-based approaches demonstrated poor performance for the inverse simulations, and shortening the time span while maintaining a large group of inliers could be a potential solution. Implementing various time spans could help identify the optimal duration for simulations. For example, determining the time span that yields the best clustering performance in inverse runs could guide a subsequent simulation. In conclusion, varying the simulation settings will likely provide valuable insights into where data reduction through clustering has the most potential.

5.2 Conclusion

The research concludes that the effectiveness of data reduction through clustering depends on the specific requirements of the analysis. Density-based methods like OPTICS and HDBSCAN are highly effective in reducing data variability but may exclude significant trajectories, potentially leading to incomplete or inaccurate results in critical applications. Distance-based methods, although less effective in data reduction, ensure that all data points are considered, making them more suitable for tasks requiring comprehensive analysis. The study highlights the need for further research to implement multiple performance measures and explore additional clustering techniques to improve the reliability and applicability of data reduction methods in maritime pollution analysis.

6. Formula list

The formula for the standard deviation σ:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{6.1}$$

where:

- N is the number of data points,
- x_i represents each data point,
- and μ is the mean of the data points.

The formula for the average standard deviation across all trajectories $\bar{\sigma}$:

$$\bar{\sigma} = \frac{1}{T}\sum_{j=1}^{T}\sigma_j \tag{6.2}$$

where:

- T is the total number of trajectories,
- σ_j is the standard deviation of the j-th trajectory, calculated with Formula 6.1.

The formula for the average standard deviation across clusters $\bar{\sigma}^{\text{weighted}}_{\text{clusters}}$:

$$\bar{\sigma}^{\text{weighted}}_{\text{clusters}} = \frac{1}{\sum_{k\in\mathbf{U},k\neq-1}|C_k|}\sum_{k\in\mathbf{U},k\neq-1}\sum_{j\in C_k}\sigma_j \tag{6.3}$$

where:

- $\mathbf{U}$ is the set of unique clusters,
- C_k represents the k-th cluster,
- $|C_k|$ is the number of trajectories in cluster C_k,
- σ_j is the standard deviation of the j-th trajectory in cluster C_k, calculated with formula 6.1,

- $\sum_{k \in \mathbf{U}, k \neq -1} |C_k|$ is the total number of trajectories across all clusters.

The formula for the weighted average standard deviation across clusters $\bar{\sigma}_{\text{cluster}}$:

$$\bar{\sigma}_{\text{weighted}} = \frac{\sum_{k \in \mathbf{U}, k \neq -1} (\bar{\sigma}_k \cdot |C_k|)}{N_{\text{total}}} \tag{6.4}$$

where:

- $\mathbf{U}$ is the set of unique clusters,
- $|C_k|$ is the number of points in cluster C_k,
- $\bar{\sigma}_k$ is the mean standard deviation for cluster k, using formula 6.2,
- N_{total} is the total count of points across all clusters, defined as $N_{\text{total}} = \sum_{k \in \mathbf{U}, k \neq -1} N_k$.

The formula for the average absolute difference MAD:

$$MAD = \frac{1}{A} \sum_{i=1}^{A} |x_i - x_{i-1}| \tag{6.5}$$

where:

- A is the total number of differences calculated,
- x_i represents the i-th data point in a trajectory,
- x_{i-1} represents the $(i-1)$-th data point in a trajectory,
- $|x_i - x_{i-1}|$ is the absolute difference between consecutive data points.

The formula for the average absolute difference within clusters MAD_{cluster}:

$$MAD_{\text{cluster}} = \frac{1}{G} \sum_{k=1}^{G} \bar{\Delta}_k \tag{6.6}$$

where:

- G is the number of clusters,
- MAD_k is the average absolute difference for the k-th cluster, calculated with formula 6.5.

The formula for the weighted average absolute difference within clusters $MAD_{\text{cluster}}^{\text{weighted}}$:

$$MAD_{\text{cluster}}^{\text{weighted}} = \frac{\sum_{k=1}^{G} (MAD_k \cdot T_k)}{T_{\text{total}}} \tag{6.7}$$

where:

- G is the number of clusters,
- MAD_k is the average absolute difference for the k-th cluster, calculated with Formula 6.5,
- T_k is the number of trajectories in cluster k,
- T_{total} is the total number of trajectories across all clusters.

The formula for the reduction in variability based on the average standard deviation (R_V):

$$R_V = \left(1 - \frac{\sigma_{\text{avg}}}{\sigma}\right) * 100 \tag{6.8}$$

where:

- σ_{avg} is the average standard deviation within clusters, calculated using formula 6.3,
- σ is the standard deviation before clustering, calculated using Formula 6.2.

The formula for the reduction in variability based on the weighted average standard deviation (R_V^{weighted}):

$$R_V^{\text{weighted}} = \left(1 - \frac{\sigma_{\text{avg}}^{\text{weighted}}}{\sigma}\right) * 100 \tag{6.9}$$

where:

- $\sigma_{\text{avg}}^{\text{weighted}}$ is the weighted average standard deviation within clusters, calculated using Formula 6.4,
- σ is the standard deviation before clustering, calculated using Formula 6.2.

The formula for the reduction in variability based on the average mean absolute difference (R_{MAD}):

$$R_{\text{MAD}} = \left(1 - \frac{MAD_{\text{avg}}}{MAD}\right) * 100 \tag{6.10}$$

where:

- MAD_{avg} is the average mean absolute difference within clusters, calculated using Formula 6.6,
- MAD is the mean absolute difference before clustering, calculated using Formula 6.5.

The formula for the reduction in variability based on the weighted mean absolute difference ($R_{\text{MAD}}^{\text{weighted}}$):

$$R_{\text{MAD}}^{\text{weighted}} = \left(1 - \frac{MAD_{\text{avg}}^{\text{weighted}}}{MAD}\right) * 100 \tag{6.11}$$

where:

- $MAD_{\text{avg}}^{\text{weighted}}$ is the weighted average mean absolute difference within clusters, calculated using Formula 6.7,
- MAD is the mean absolute difference before clustering, calculated using Formula 6.5.

7. Appendices

7.1 Appendix A - Data exploration

Table 7.1: Exploration of attributes from AIS data.

Attribute	Number of unique values	Number of NULL	Data type
MMSI	1029	0	Integer
IMO	879	0	Integer
Vessel name	1019	0	Object
Vessel type	85	0	Object
Length (m)	643	0	Float
Flag	51	276	Object
Status	15	0	Integer
Speed (knots x10)	327	38	Float
Longitude	104631	0	Float
Latitude	53278	0	Float
Course	360	7649	Float
Heading	360	36432	Float
Timestamp (UTC)	126757	0	Object

7.2 Appendix B - Simulations

Table 7.2: Input parameters for the wind model (Breivik et al., 2011).

Parameter	Value in cm/s
Downwind	0.011
Standard deviation downwind	0.031
Crosswind - Left	-0.0062
Standard deviation crosswind - Left	0.046
Crosswind - Right	0.0086
Standard deviation crosswind - Right	0.041

Table 7.3: Statistics of simulated trajectories.

Simulation type	$\sigma_{longitude}$	$\sigma_{latitude}$	$MAD_{longitude}$	$MAD_{latitude}$
Forward *STELLA ORION*	0.065	0.051	0.002	0.001
Forward *SANTOS EXPRESS*	0.068	0.055	0.002	0.001
Inverse Barrel 4	0.126	0.032	0.002	0.001
Inverse Barrel 9	0.195	0.045	0.002	0.001

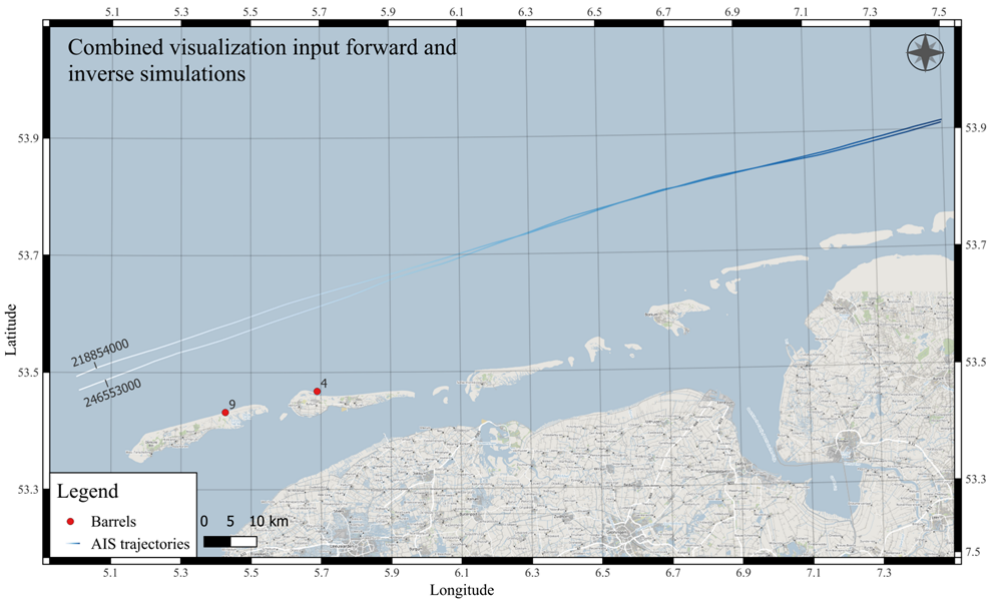
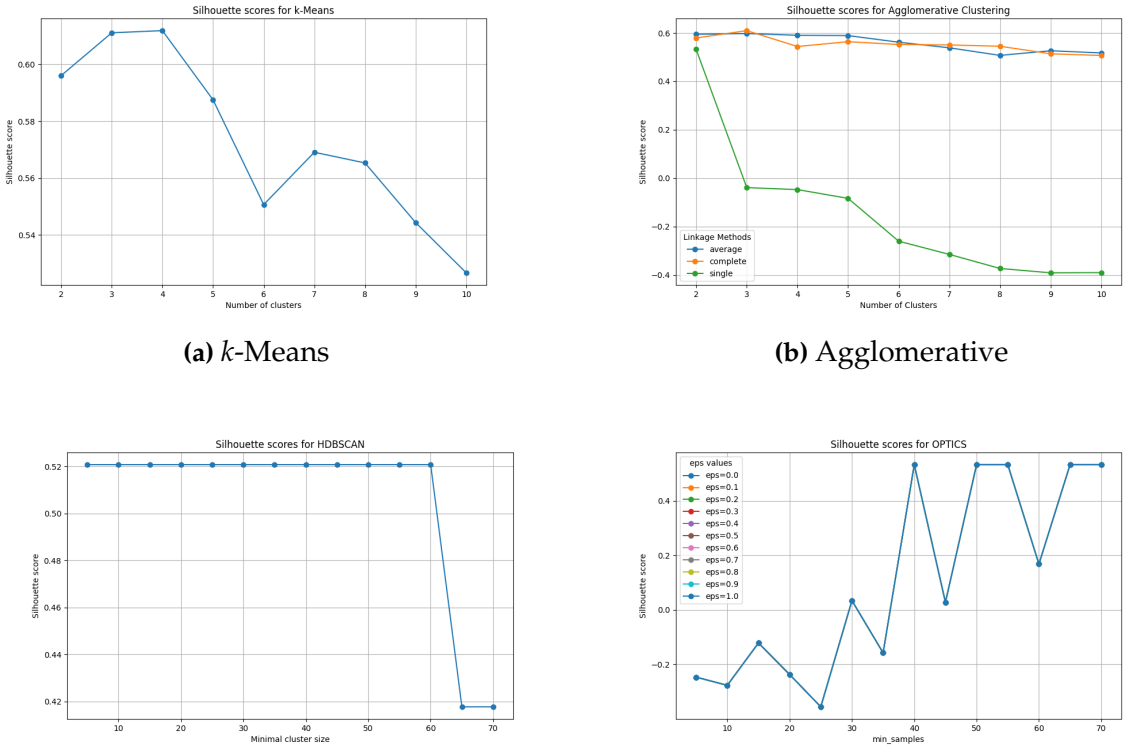


Figure 7.1: AIS trajectories of ships *STELLA ORION* (MMSI: 246553000) and *SANTOS EXPRESS* (MMSI: 218854000), where the light color indicates the start of their trajectories and the dark color the end. Barrels 4 and 9 were displayed as red dots.

7.3 Appendix C - Silhouette scores



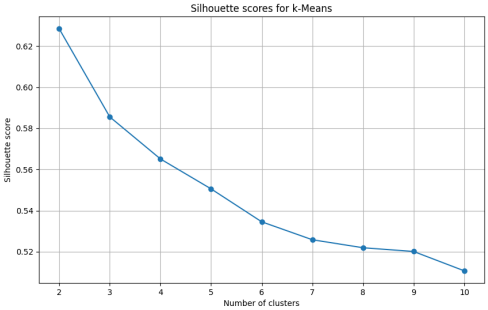
(a) *k*-Means

(b) Agglomerative

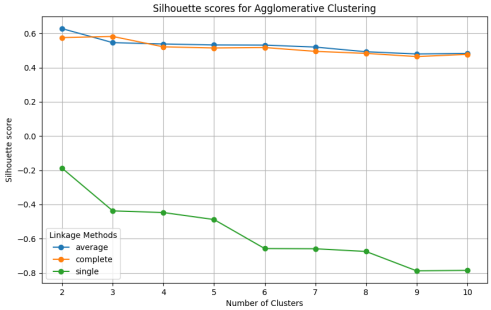
(c) HDBSCAN

(d) OPTICS

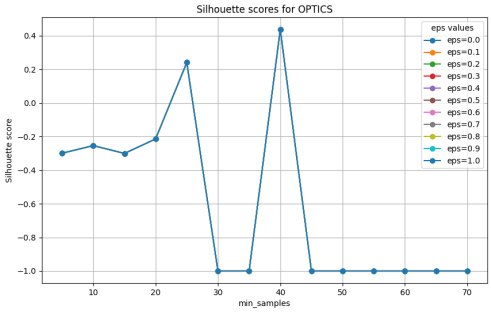
Figure 7.2: The silhouette scores with different parameter settings for the four cluster methods for *STELLA ORION*.



(a) *k*-Means

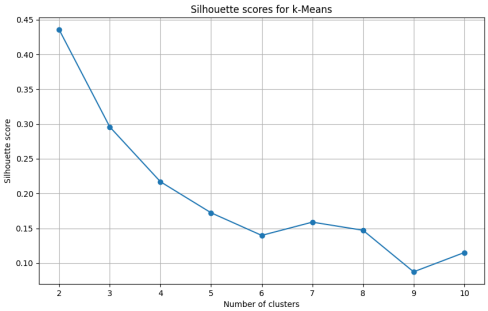


(b) Agglomerative

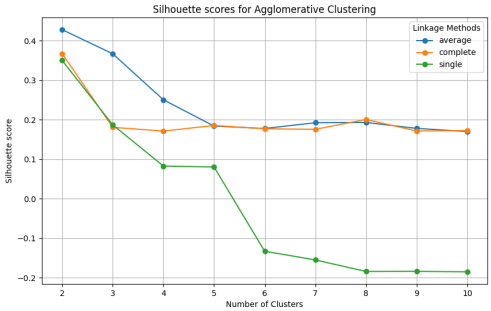


(c) OPTICS

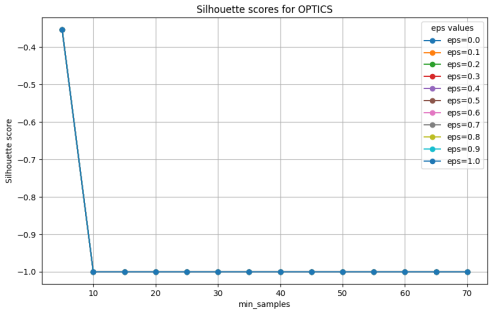
Figure 7.3: The silhouette scores with different parameter settings for the four cluster methods for *SANTOS EXPRESS*.



(a) *k*-Means

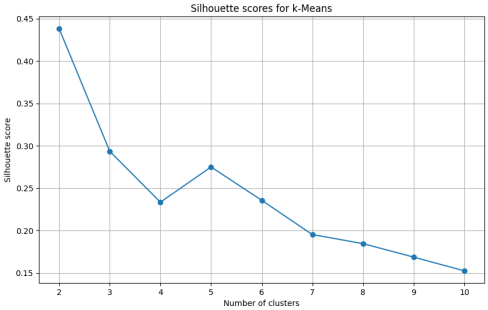


(b) Agglomerative

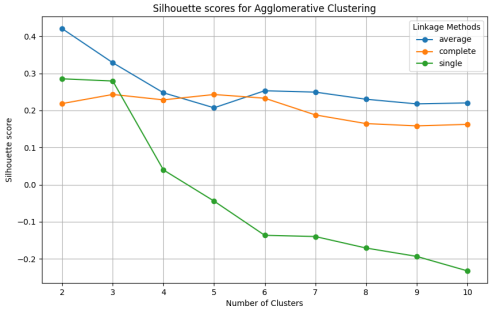


(c) OPTICS

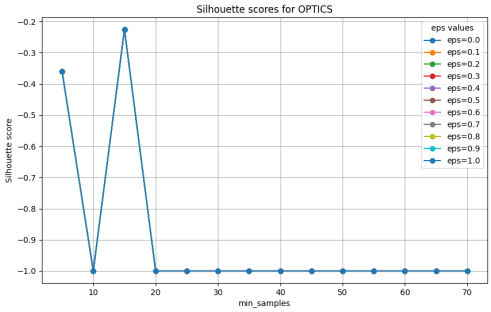
Figure 7.4: The silhouette scores with different parameter settings for the four cluster methods for Barrel 4.



(a) *k*-Means



(b) Agglomerative



(c) OPTICS

Figure 7.5: The silhouette scores with different parameter settings for the four cluster methods for Barrel 9.

Table 7.4: Overview of the highest silhouette scores.

Simulation	Cluster method	Silhouette score
Forward *STELLA ORION*	*k*-Means	0.612
	Agglomerative	0.610
	HDBSCAN	0.532
	OPTICS	0.532
Forward *SANTOS EXPRESS*	*k*-Means	0.628
	Agglomerative	0.628
	HDBSCAN	0.318
	OPTICS	0.437
Inverse Barrel 4	*k*-Means	0.436
	Agglomerative	0.428
	HDBSCAN	0.09
	OPTICS	-0.353
Inverse Barrel 9	*k*-Means	0.438
	Agglomerative	0.421
	HDBSCAN	0.052
	OPTICS	-0.226

7.4 Appendix D - Cluster results

Table 7.5: Average standard deviation and MAD within clusters.

Simulation	Cluster method	$\bar{\sigma}_{\text{cluster}}$ (%)		*MAD*	
		Longitude	**Latitude**	**Longitude**	**Latitude**
Forward *STELLA ORION*	*k*-Means	0.065	0.051	0.002	0.001
	Agglomerative	0.065	0.051	0.002	0.001
	HDBSCAN	0.065	0.051	0.002	0.001
	OPTICS	0.065	0.051	0.002	0.001
Forward *SANTOS EXPRESS*	*k*-Means	0.068	0.055	0.002	0.001
	Agglomerative	0.068	0.055	0.002	0.001
	HDBSCAN	0.070	0.055	0.002	0.001
	OPTICS	0.069	0.055	0.002	0.001
Inverse Barrel 4	*k*-Means	0.126	0.032	0.002	0.001
	Agglomerative	0.126	0.032	0.002	0.001
	HDBSCAN	0.126	0.030	0.002	0.001
	OPTICS	0.147	0.030	0.002	0.001
Inverse Barrel 9	*k*-Means	0.195	0.045	0.002	0.001
	Agglomerative	0.195	0.045	0.002	0.001
	HDBSCAN	0.194	0.043	0.002	0.001
	OPTICS	0.187	0.041	0.002	0.001

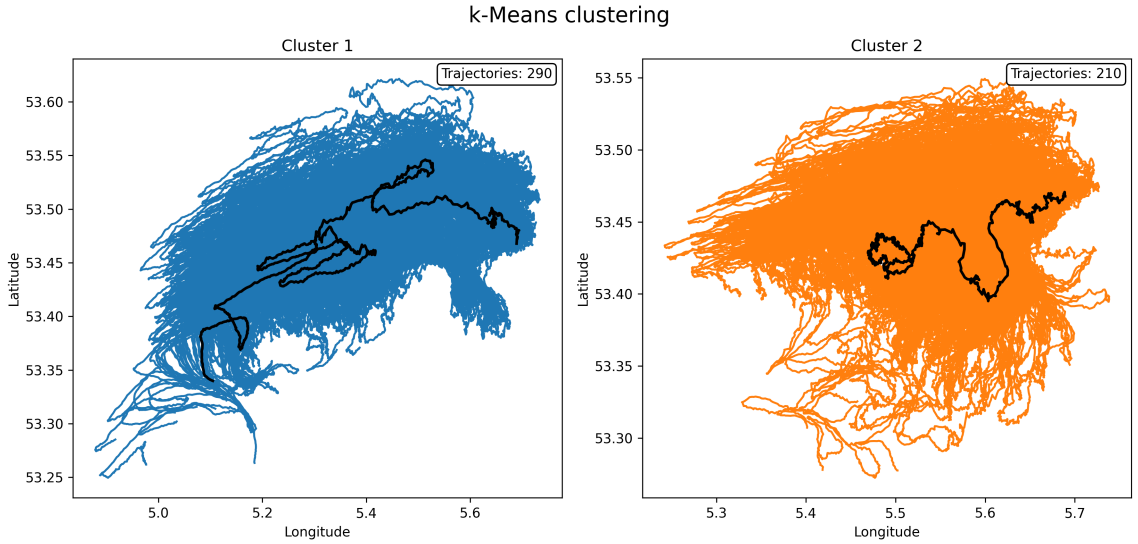


Figure 7.6: The clustering result of Barrel 4 using *k*-means. The medoid of each cluster is visualized as a black line.

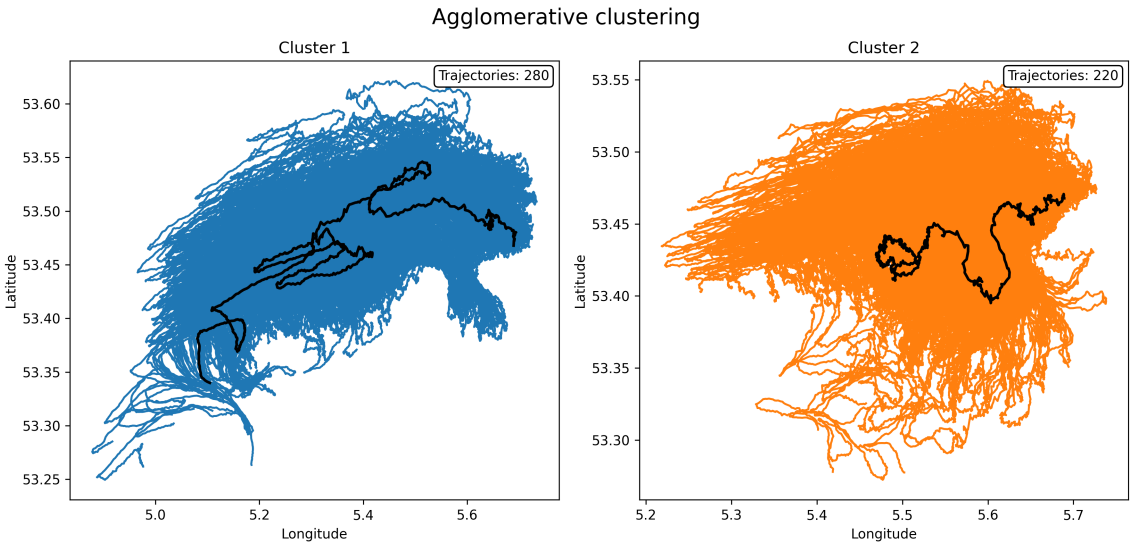


Figure 7.7: The clustering result of Barrel 4 using agglomerative clustering. The medoid of each cluster is visualized as a black line.

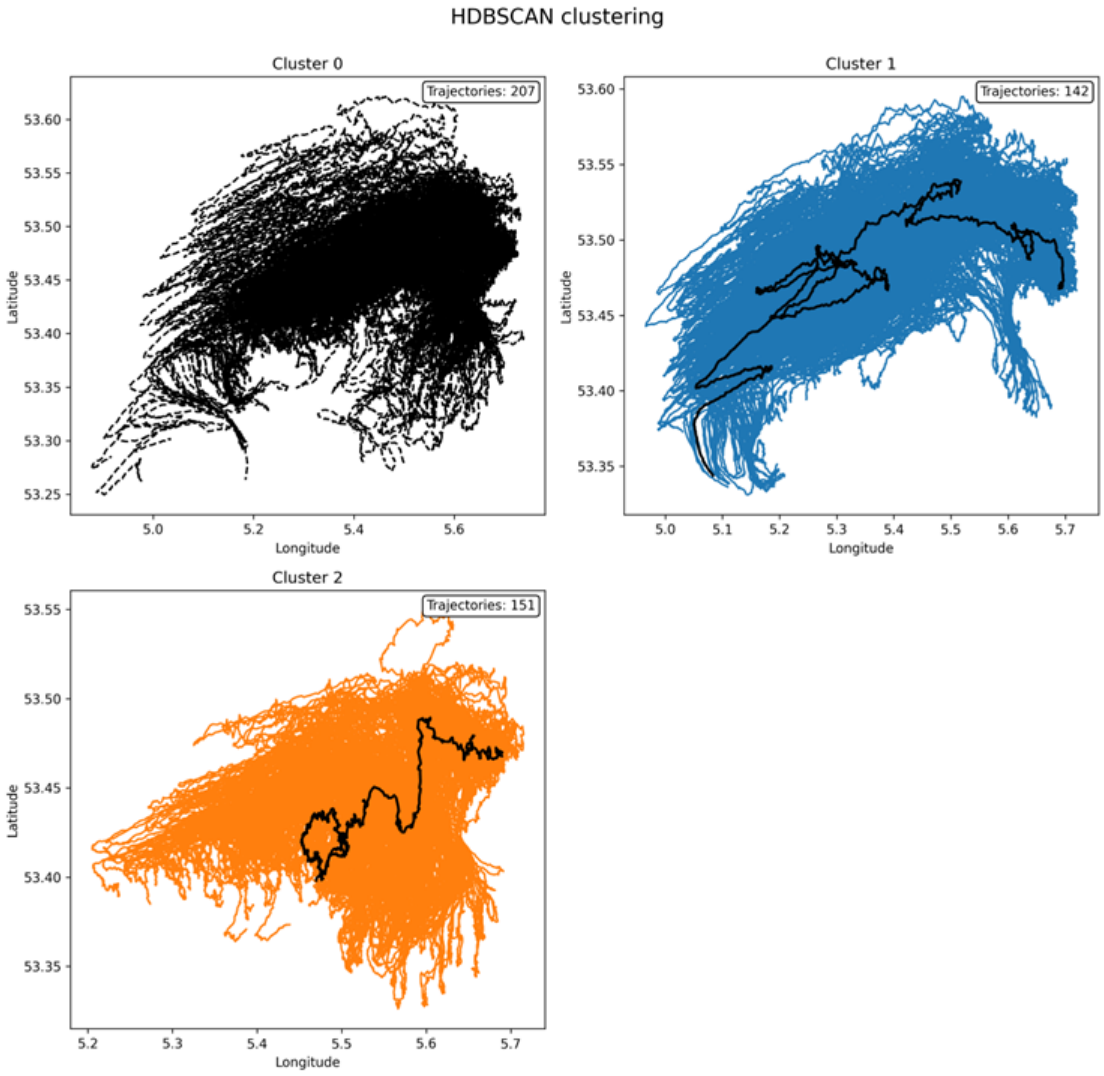


Figure 7.8: The clustering result of Barrel 4 using HDBSCAN. The medoid of each cluster is visualized as a black line, with cluster 0 containing the noise.

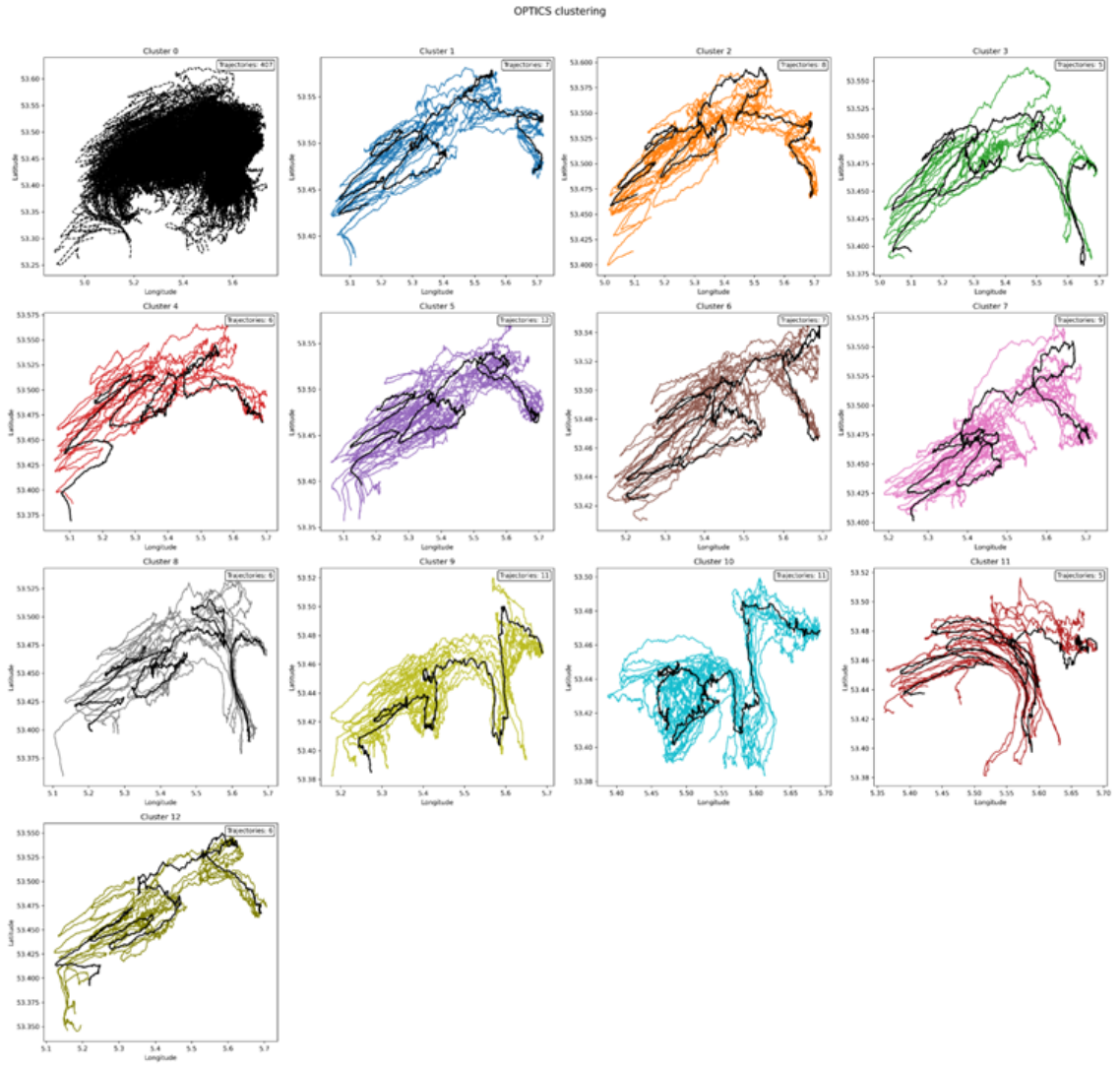


Figure 7.9: The clustering result of Barrel 4 using OPTICS. The medoid of each cluster is visualized as a black line, with cluster 0 containing the noise.

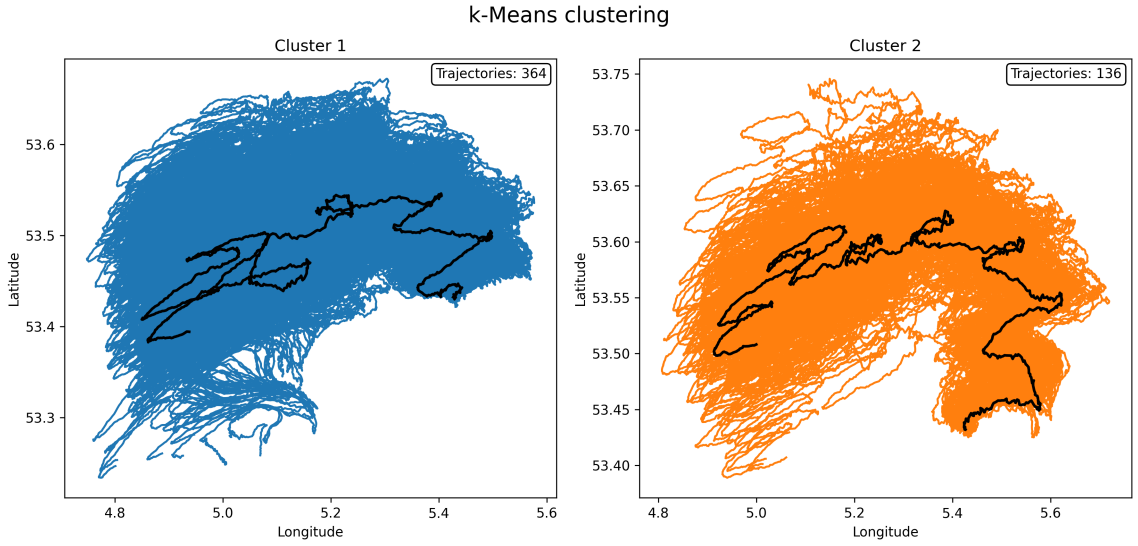


Figure 7.10: The clustering result of Barrel 9 using *k*-means. The medoid of each cluster is visualized as a black line.

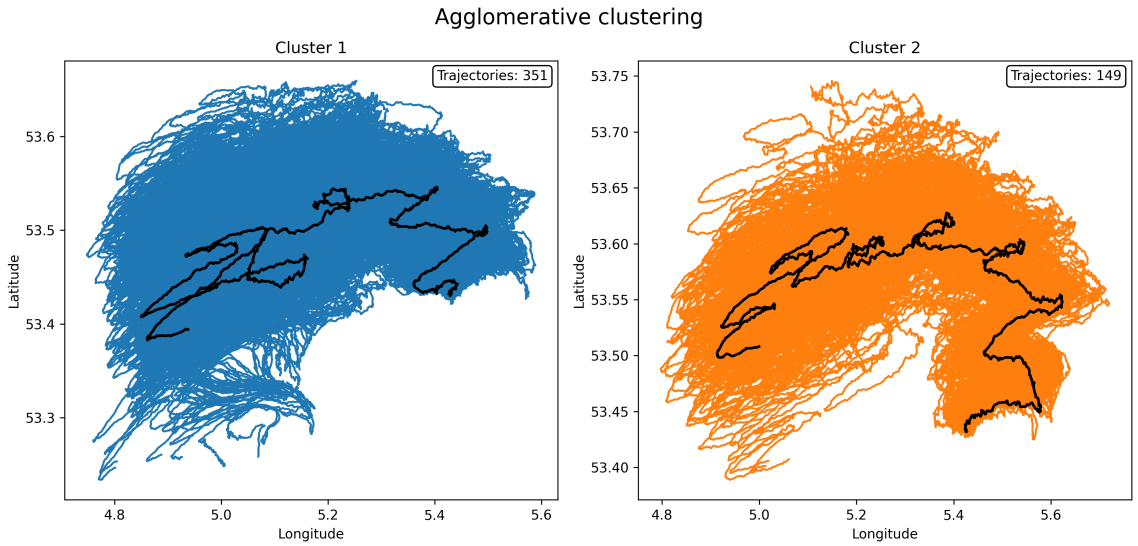


Figure 7.11: The clustering result of Barrel 9 using agglomerative clustering. The medoid of each cluster is visualized as a black line.

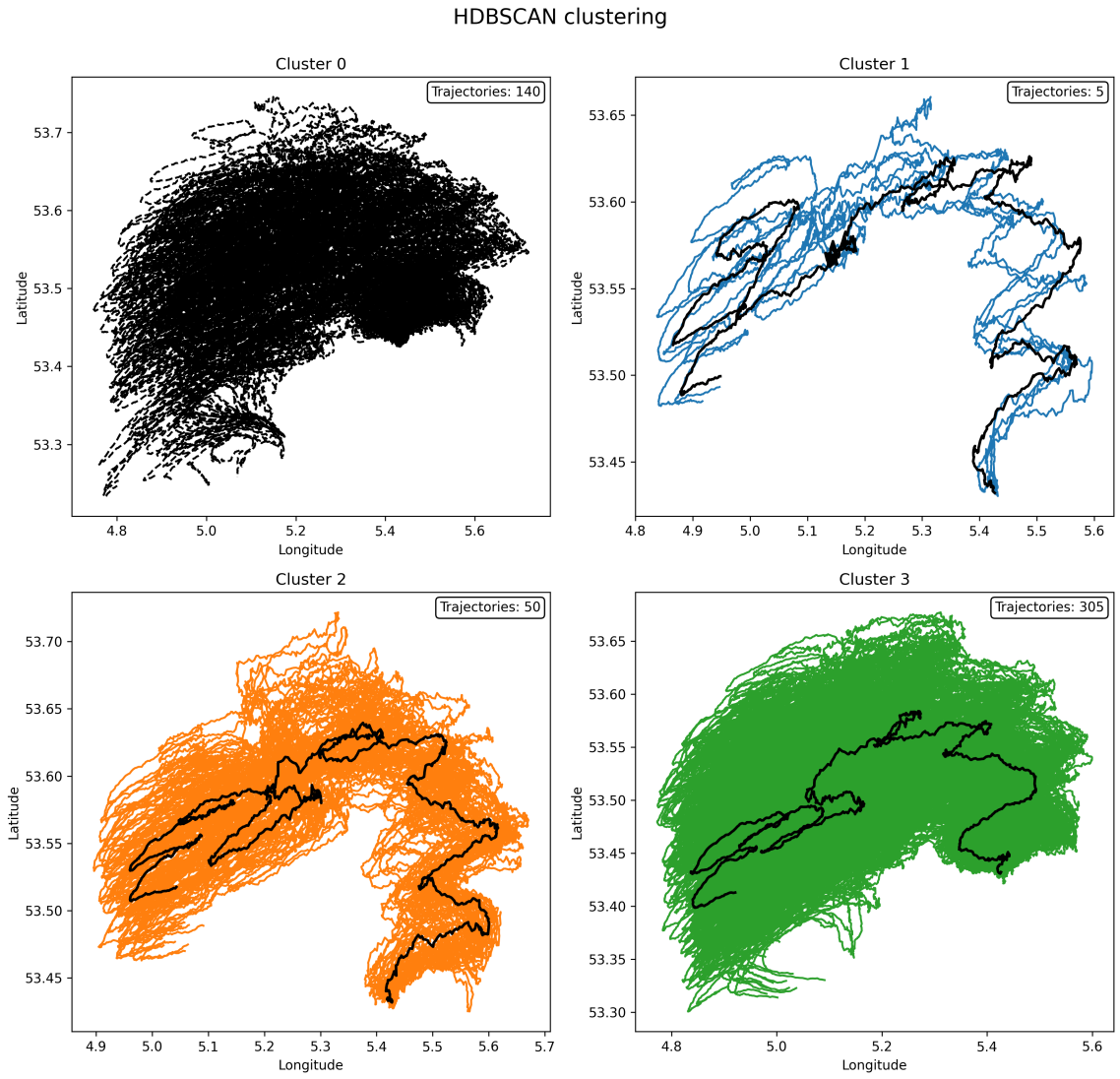


Figure 7.12: The clustering result of Barrel 9 using HDBSCAN. The medoid of each cluster is visualized as a black line, with cluster 0 containing the noise.

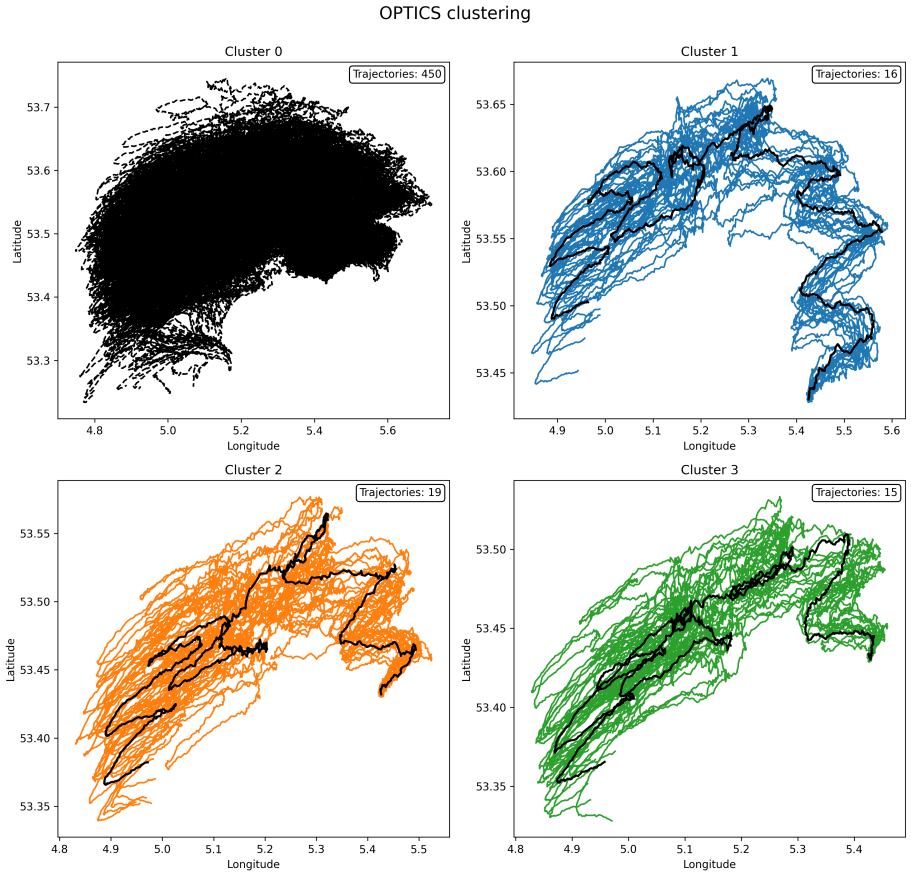


Figure 7.13: The clustering result of Barrel 9 using OPTICS. The medoid of each cluster is visualized as a black line, with cluster 0 containing the noise.

Bibliography

Advances in knowledge discovery and data mining. (2013). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-37456-2

Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, *9*(8). https://doi.org/10.3390/electronics9081295

Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). Optics: Ordering points to identify the clustering structure. *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, 49–60. https://doi.org/10.1145/304182.304187

Bascom, W. (1974). The disposal of waste in the ocean. *Scientific American*, *231*(2), 16–25.

Bouveyron, C., & Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics Data Analysis*, *71*, 52–78. https://doi.org/https://doi.org/10.1016/j.csda.2012.12.008

Breivik, Ø., Allen, A. A., Maisondieu, C., & Roth, J. C. (2011). Wind-induced drift of objects at sea: The leeway field method. *Applied Ocean Research*, *33*(2), 100–109. https://doi.org/10.1016/j.apor.2011.01.005

BRYAN, K. (1969). Climate and the ocean circulation: Iii. the ocean model. *Monthly Weather Review*, *97*(11), 806–827. https://doi.org/10.1175/1520-0493(1969)097<0806:CATOC>2.3.CO;2

Deltares. (2024). https://www.deltares.nl/en

Estivill-Castrol, V., & Murray, A. T. (1998). Discovering associations in spatial data — an efficient medoid based approach. In X. Wu, R. Kotagiri, & K. B. Korb (Eds.), *Research and development in knowledge discovery and data mining* (pp. 110–121). Springer Berlin Heidelberg.

Giannotti, F., Giannotti, G., & Pedreschi, D. (2008, January). *Mobility, data mining and privacy: Geographic knowledge discovery*. https://doi.org/10.1007/978-3-540-75177-9

Gudmundsson, J., & van Kreveld, M. (2006). Computing longest duration flocks in trajectory data. *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*. https://doi.org/10.1145/1183471.1183479

IMO. (2024). https://www.imo.org/en

Jimoh, B., Mariescu-Istodor, R., & Fränti, P. (2022). Is medoid suitable for averaging gps trajectories? *ISPRS International Journal of Geo-Information*, *11*(2). https://doi.org/10.3390/ijgi11020133

Kanagala, H. K., & Jaya Rama Krishnaiah, V. (2016). A comparative study of k-means, dbscan and optics. *2016 International Conference on Computer Communication and Informatics (ICCCI)*, 1–6. https://doi.org/10.1109/ICCCI.2016.7479923

KNMI. (2024). https://www.knmi.nl/home

Kpler. (2024). https://www.kpler.com/

kpler. (n.d.-a). SANTOS EXPRESS. https://www.marinetraffic.com/en/ais/details/ships/shipid:4939224/mmsi:218854000/imo:9777632/vessel:SANTOS_EXPRESS

kpler. (n.d.-b). STELLA ORION. https://www.marinetraffic.com/en/ais/details/ships/shipid:269386/mmsi:246553000/imo:9265251/vessel:STELLA_ORION

Krislock, N., & Wolkowicz, H. (2012). Euclidean distance matrices and applications. In M. F. Anjos & J. B. Lasserre (Eds.), *Handbook on semidefinite, conic and polynomial optimization* (pp. 879–914). Springer US. https://doi.org/10.1007/978-1-4614-0769-0_30

Marine Digital. (2024). Automatic Identification System (AIS). What is AIS Data? https://marine-digital.com/article_ais

MMSI. (2022, June). https://www.fcc.gov/wireless/bureau-divisions/mobility-division/maritime-mobile/ship-radio-stations/maritime-mobile#:~:text=Maritime%20Mobile%20Service%20Identities%20(MMSIs,or%20a%20coast%20radio%20station.

Murtagh, F., & Contreras, P. (2011). Algorithms for hierarchical clustering: An overview. *WIREs Data Mining and Knowledge Discovery*, *2*(1), 86–97. https://doi.org/10.1002/widm.53

Nanni, M., & Pedreschi, D. (2006). Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, *27*(3), 267–289. https://doi.org/10.1007/s10844-006-9953-7

Pastor, D., & Socheleau, F.-X. (2012). Robust estimation of noise standard deviation in presence of signals with unknown distributions and occurrences. *IEEE Transactions on Signal Processing*, *60*(4), 1545–1555. https://doi.org/10.1109/TSP.2012.2184534

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Petrovic, S. (2006). A comparison between the silhouette index and the davies-bouldin index in labelling ids clusters. *Proceedings of the 11th Nordic workshop of secure IT systems*, *2006*, 53–64.

Politiebureau Terschelling. (2024). https://www.politie.nl/mijn-buurt/politiebureaus/01/fryslan/terschelling.html

Rahman, M. F., Liu, W., Suhaim, S. B., Thirumuruganathan, S., Zhang, N., & Das, G. (2016). Hdbscan: Density based clustering over location based services.

Rezaie, M., & Saunier, N. (2021). Trajectory clustering performance evaluation: If we know the answer, it's not clustering. https://doi.org/10.48550/ARXIV.2112.01570

Rokach, L., & Maimon, O. (2005). Clustering methods. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 321–352). Springer US. https://doi.org/10.1007/0-387-25465-X_15

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

RWS. (2024). https://www.rijkswaterstaat.nl/

RWS & Deltares. (2022). *D-Flow FM 3D Noordzee* (tech. rep.).

Santos, J. M., & Embrechts, M. (2009). On the use of the adjusted rand index as a metric for evaluating supervised classification. In C. Alippi, M. Polycarpou, C. Panayiotou, & G. Ellinas (Eds.), *Artificial neural networks – icann 2009* (pp. 175–184). Springer Berlin Heidelberg.

Shutaywi, M., & Kachouie, N. N. (2021). Silhouette analysis for performance evaluation in machine learning with applications to clustering. *Entropy*, *23*(6). https://doi.org/10.3390/e23060759

Song, M.-H., Kang, E.-S., Jeong, C.-H., Chow, M.-Y., & Ayhan, B. (2003). Mean absolute difference approach for induction motor broken rotor bar fault detection. *4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, 2003. SDEMPED 2003.*, 115–118. https://doi.org/10.1109/DEMPED.2003.1234557

Sterl, A., & Ministry of Infrastructure and Water Management. (2019, July). *Wave-dependent drag parametrizations and their impact on drag and water levels* (tech. rep. No. TR-374). Ministry of Infrastructure; Water Management. https://cdn.knmi.nl/knmi/pdf/bibliotheek/knmipubTR/TR374.pdf

Stewart, G., & Al-Khassaweneh, M. (2022). An implementation of the hdbscan* clustering algorithm. *Applied Sciences*, *12*(5). https://doi.org/10.3390/app12052405

The SWAN team: SWAN - Scientific and technical documentation SWAN Cycle III version 41.20A. (n.d.). https://swanmodel.sourceforge.io/online_doc/swantech/swantech.html

Van der Minnen, M. (2024, June). *Employ Drift Modeling and Maritime Traffic Analysis to Identify Sources of Marine Oil Barrel Pollution* (tech. rep.).

Wai, K. P., & Nwe, N. (2017). Measuring the distance of moving objects from big trajectory data. *International Journal of Networked and Distributed Computing*, *5*(2), 113. https://doi.org/10.2991/ijndc.2017.5.2.6

Wang, X., & Xu, Y. (2019). An improved index for clustering validation based on silhouette index and calinski-harabasz index. *IOP Conference Series: Materials Science and Engineering*, *569*(5), 052024. https://doi.org/10.1088/1757-899X/569/5/052024

Yim, O., & Ramdeen, K. T. (2015). Hierarchical cluster analysis: Comparison of three linkage measures and application to psychological data. *The Quantitative Methods for Psychology*, *11*(1), 8–21. https://doi.org/10.20982/tqmp.11.1.p008