

Automated port discovery using AIS ship tracking data



Tomas Stankevičius, Julija Raudytė, Kristina Kazlauskaitė

2025

Task overview

- **Dataset:** AIS ship tracking data from Baltic Sea region (February 14, 2025)
- **Goal:** automatically identify port locations and create a comprehensive map of active ports based on real ship behavior data

Port detection algorithm overview

1. Stop detection

- Minimum stop duration: **1 hour**
- Maximum duration between pings: **15 minutes**
- Per-ship (MMSI) parallel analysis

2. DBSCAN spatial clustering

- **1 km radius** for port boundaries (Haversine distance)
- **7 stops** required to form a cluster
- Noise filtering (DBSCAN labels noise as -1)

3. Port boundary generation

- Each cluster is processed independently
- Convex hull algorithm creates polygon boundaries

Implementation

- `parallel_data_processing.py` – AIS data preprocessing:
 - Parallel chunk processing (1M rows per chunk)
 - Removes NaNs
 - Filter stationary vessels ($\text{SOG} \leq 2$ knots)
- `clustering_and_port_detection.py` – analysis:
 - Identifies significant stop events in parallel
 - Clusters stop events using DBSCAN based on their geographical coordinates
- `parallel_polygon_processing.py` – geometric processing
 - Calculates the convex hull WKT for a single cluster's stops
 - Parallel processing of port boundaries
- `utils.py` – visualization and output
 - Generates port map

Results

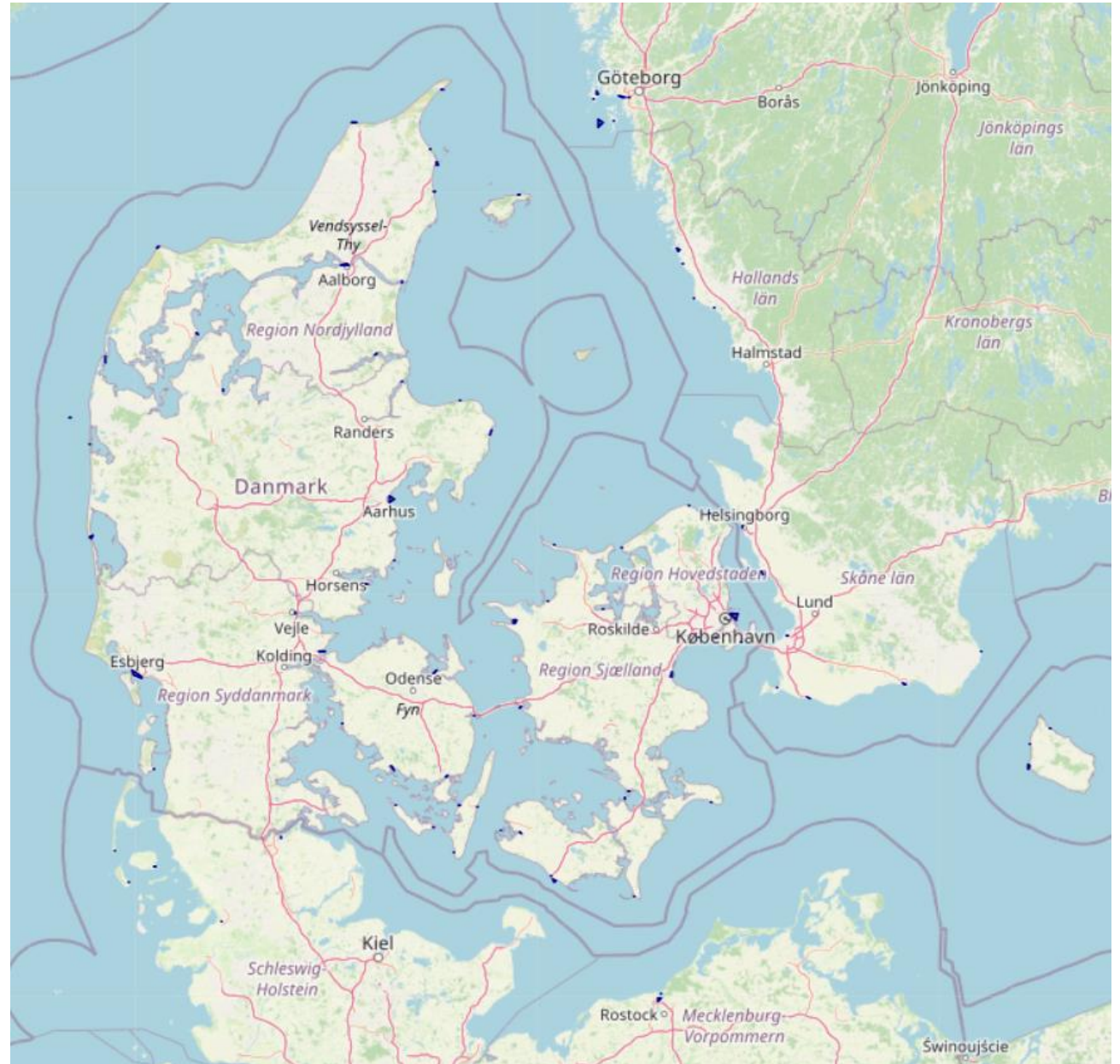
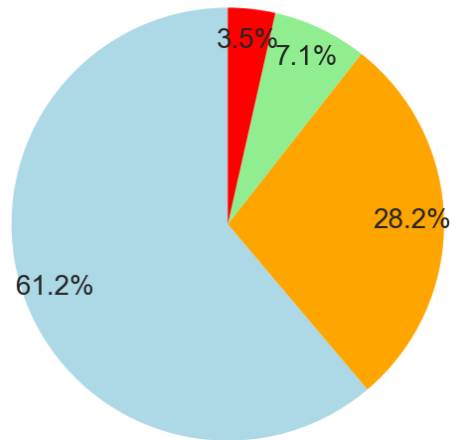
- Largest identified ports (based on the number of ships):

Cluster ID	Unique Ships	Total Stops	Avg Duration (hrs)	Port Name (based on coordinates)
48	82	97	15.4	Port of Hirthals Scandinavia
0	79	105	15.5	Esbjerg Port
36	74	93	15.4	Port of Skagen
22	43	59	13.6	Thyboron Port
24	38	41	19.4	Port of Hanstholm

Port map

- 85 ports identified in total

Port Distribution by Size
(Based on Number of Ships)



Conclusions

- AIS ship tracking **generates large volumes of data** at high speed, requiring adaptive tools capable of processing big data efficiently.
- In this work, **parallel computing** techniques in Python were employed to load, preprocess, and analyze one day of AIS data for port detection, which allowed efficient use of computing resources and **reducing analysis time**.
- As a result of the analysis, **85 ports were detected**, majority of which were relatively small (up to 10 ships) and only 3 hosted over 50 ships.