Automated port discovery using AIS ship tracking data



Tomas Stankevičius, Julija Raudytė, Kristina Kazlauskaitė

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 (SOG ≤ 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):

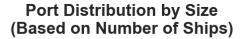
Cluster ID	Unique Ships	Total Stops	Avg Duration (hrs)	Port Name (based on coordinates)
48	82	97	15.4	Port of Hirthals Scandinavia
0	79	105	15.5	Esbjerg Port
36	74	93	15.4	Port of Skagen
22	43	59	13.6	Thyboron Port
24	38	41	19.4	Port of Hanstholm

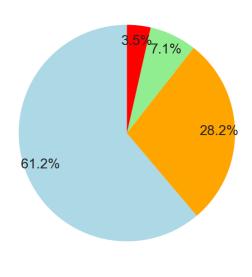
Port map

Small (1-10): 52 ports

Medium (11-25): 24 ports

• 85 ports identified in total





Large (26-50): 6 ports

Major (50+): 3 ports

