GPS Spoofing in Maritime Navigation

Danial Yntykbay

March 27, 2025

Outline

1. Methods to detect anomalies
2. Parallelization
3. Results
4. Results

Speed and Location anomalies

```python
def speed_in(group, max_distance=10, difference_speed=100):
    # Some processing here...

    # Check distance anomaly
    if d > max_distance:
        anomalies.append(...)

    # Check speed anomaly
    if abs(calculated_speed - reported_speed) >
   difference_speed:
        speed_anomalies.append(...)

```

Conflicting Positions

```python
def neighbor_vessels(chunk, conflicting_distance=0.2):
    # Some processing here...

    if dist < conflicting_distance:
        anomalies.append(...)

```

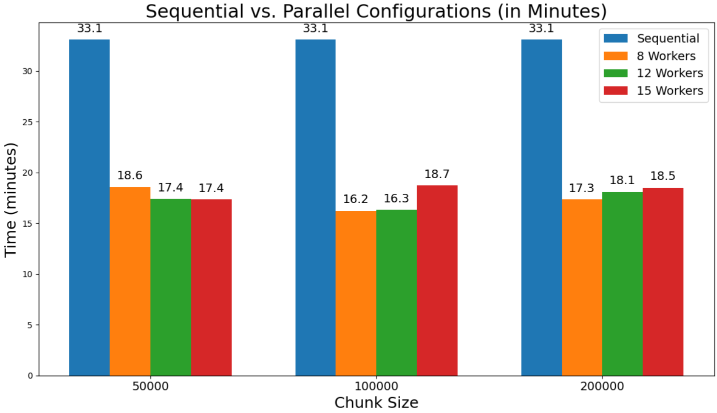
Parallelizing

```python
def parallelize_MMSI(file, chunk_size=100000, num_workers=None):
    # Some processing here...

    pool = mp.Pool(num_workers)
    for chunk in chunks:
        groups = [group for _, group in chunk.groupby("MMSI")]
        results = pool.map(speed_in, groups)

```

Parallelizing

```python
def parallelize_gps(file, chunk_size=10000):
    # Some processing here...

    for chunk in chunks:

    groups = [group for _, group in chunk.groupby(pd.Grouper(key=’# Timestamp’, freq=’5min’))]

    # Process groups in parallel
    results = pool.map(neighbor_vessels, groups)

```

Results


Sequential vs. Parallel Configurations (in Minutes)
Time (minutes)
Chunk Size
0
5
10
15
20
25
30
50000
100000
200000
Sequential
8 Workers
12 Workers
15 Workers
33.1
18.6
17.4
17.4
33.1
16.2
16.3
18.7
33.1
17.3
18.1
18.5