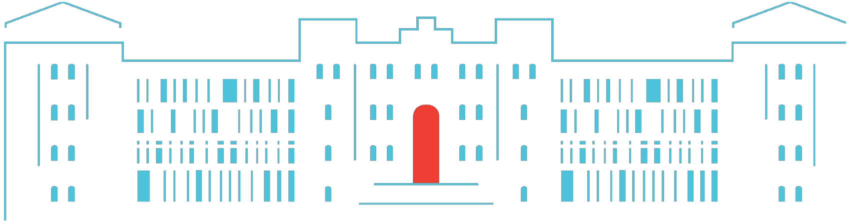
ML in maritime logistics: Detecting abnormal trajectories of vessels

TUHH
Technische Universität Hamburg

16.07.2025



Veronika Anokhina, Piotr Ciupiński, Rafał Mironko

Problem description

TUHH

- Detection of abnormal trajectories can be useful to detect and avoid collision or any misbehavior in the sea.

- There are ships using two routes: Bremerhaven-Hamburg and Kiel-Gdynia.

- Every few minutes, each ships sends a signal with current state.

- We need to create a tool able to look at any historical trip and classify each signal: if there is an anomaly or not.

- An anomaly may be caused by:
  - Choosing a suboptimal route
  - Stopping for a long time
  - Drastically changing speed
  - Drastically changing course
  - Unstable draught
  - Irregular speed values

What was solved and how

TUHH

Data preparation

- A cleaning pipeline was created, normalizing names, purging clear outliers and filling missing values, wherever possible.
- Labels indicating anomalous and non-anomalous entries were added, based on our definition of anomaly.

Features used:

- Speed_over_ground
- Deltas: dv, dcourse, ddraft
- Port marking: zone
- x_km, y_km, dist_to_ref

Anomaly detection

- Four models were selected and trained:
  - Random forest
  - Logistic regression
  - Isolation forest
  - One-class support vector machine
- Long Short-Term Autoencoder Neural Network was also implemented to try detecting anomalies (mandatory).
- A django-vue web application was created to visualize and analyze results obtained by models.

Quality of solution Pt.1

TUHH

F1-score was selected as primary evaluation metric:

- Considers both precision (avoiding false alarms and recall (detecting actual anomalies) equally important.
- Prevents models from simply predicting the majority class (normal) to achieve high accuracy.
- Easily interpretable - higher F1-score usually means better overall performance of a model.

A fraction of labeled data was separated to create a test set. After its evaluation by the models, they obtained following scores(route 1/route 2):

- Random forest: F1 = 0.992/0.972
- One-class SVM: F1 = 0.941/0.900
- Isolation forest: F1 = 0.884/0.875
- Logistic regression: F1 = 0.859/0.760

Quality of solution Pt.2

TUHH

LSTM Autoencoder:

- **Learns**
  to reconstruct normal time series through an encoder-decoder architecture; high reconstruction error indicates anomaly.
- **Captures**
  temporal dependencies in sequential data as trajectories
- **Assumes**
  anomalies significantly deviate from learned normal temporal patterns in reconstruction quality.

Observations during training:

- **Performance scores achieved:**
  A validation set was created from labeled data and the model was evaluated with the following results:

  Route 1 (Kiel): F1 = 0.879
  Route 2 (Bremerhaven): F1 = 0.807

- **However**, since the model evaluates entire route snippets rather than individual points, these metrics don't fully reflect actual performance on our data. The limited validation data and the method used for it wasn't extensive enough to create sufficient examples for proper evaluation, and it was discovered too late in the development

TUHH

DEMO

Evaluation of development process

Our used workflow:

- Had longer meetings when approaching major milestone in the project
- Often held shorter meeting talking about everyone's progress
- Consistent meeting schedule
- Detailed evaluation before each sprint

Possible improvements:

- Having a defined team leader who organizes tasks and supports each member a little.
- Better estimating total time necessary to finish the project, avoiding underestimating the time actually needed

QUESTIONS