UiT The Arctic University of Norway

The Faculty of Science and Technology,
Department of Computer Science

Predicting the Destination Port of Fishing Vessels

Andreas Berntsen Løvland

Master's Thesis in Computer Science - INF-3981 - June 2024

Supervisors

Main supervisor:	John Markus Bjørndalen	UiT The Arctic University of Norway, Faculty of Science and Technology, Department of Computer Science
Co-supervisor:	Helge Fredriksen	UiT The Arctic University of Norway, Faculty of Science and Technology, Department of Computer Science

Abstract

Regulating the catch of fishing vessels is crucial for maintaining sustainable fish populations, preventing illegal fishing, and ensuring the quality of the fish being delivered. One effective method of controlling the catch is to have controllers physically present at the port where the catch is being delivered. However, vessels do not always report their destination port in a timely manner, which limits the ability of controllers to regulate the delivered catch.

In order to improve the ability of controllers to regulate the catch, this thesis explores how to forecast the destination ports of fishing vessels without relying on their manually transmitted information. We utilize Automatic Identification System (AIS) data to analyze the movement patterns and behaviors of fishing vessels in our dataset, and extend existing work on vessel trajectory predictions using machine learning, to forecast the destination ports of fishing vessels. Additionally, we develop a statistical baseline model to compare our results against.

Our results demonstrate that both models correctly predict the destination port of a given vessel in the majority of times, with the accuracy of the machine learning approach increasing as more input data is added. The statistical baseline mode performs better with vessels that do not visit a variety of ports, while the machine learning approach provides a better overall assessment, and thus appears to be the more promising approach. Both models have the potential to be improved considerably by incorporating more input features.

Acknowledgements

I want to thank my supervisors, John Markus Bjørndalen and Helge Fredriksen, for their valuable guidance and insight throughout this thesis. It has been a very motivating experience.

I also want to thank Norges Råfisklag for providing the initial idea and inspiration for this project.

Lastly, I want to thank my family for their support throughout all these years.

Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

AIS Automatic Identification System

API Application Programming Interface

Bi-GRU Bi-directional GRU

Bi-LSTM Bi-directional LSTM

BP Back Propagation Network

COG Course Over Ground

CSV Comma-Separated Values

DCA Detailed Catch and Activity report

DEP Departure report

DMA Danish Maritime Authority

EEZ Exclusive Economic Zone

ERS Electronic Reporting System

EU European Union

FDE Final Displacement Error

GPR Gaussian Process Regression

GRU Gate Recurrent Unit

IMO International Maritime Organization

JSON JavaScript Object Notation

KF Kalman Filter

LSTM Long Short-Term Memory

MAE Mean Absolute Error

MMSI Maritime Mobile Service Identity

MSE Mean Squared Error

POR Port report

RF Random Forest

RNN Recurrent Neural Network

ROI Region of Interest

SAR Search and Rescue

Seq2seq Sequence to Sequence

SMAPE Symmetrical Mean Absolute Percentage Error

SOG Speed Over Ground

SVR Support Vector Regression

TRA Transhipment report

VHF Very High Frequency

VMS Vessel Monitoring System

VTS Vessel Traffic Service

1 Introduction

The fishing industry is one of Norway's most valuable industries, exporting millions of tons of fish each year to other countries. The industry is driven by fishing vessels being out in the sea catching fish, and then selling their catch. This process continues until the vessels have reached their quota, which is the limit of how much fish they are allowed to catch.

When the fishing vessels are done with the fishing for their current trip, they head to a port where they can deliver and/or sell their catch. To ensure that the catch of a vessel is within the regulations that are set by the authorities, e.g., the amount of fish they have caught or if the species caught are allowed, controllers[1] can travel to a port to control the catch of incoming fishing vessels. However, due to the large number of ports in Norway, each with various amounts of activity, it is not feasible to have controllers deployed at each one, nor all the time. Therefore, the controllers have to choose when and where they should be located when they are to perform controls, which can be based on which vessels are heading in, and where.

Today, the controllers use a combination of methods to collect information about what vessels are likely to arrive to which ports. These methods include: monitoring the GPS activity of vessels which are nearby the desired locations through Automatic Identification System (AIS) messages, monitoring the Electronic Reporting System (ERS) messages the vessels transmit, and making

1. People whose job is to control that the catch of a fishing vessel satisfies the regulations

phone calls to other companies, or ports, to figure out if they have info about what fishing vessels are heading in, when, and where. By having multiple possible information channels, it is not easy to obtain accurate information quickly, which is essential for the controllers to be able to move to the corrects ports when a vessel they find interest in to control is heading in.

The aim of this thesis is to investigate how to accurately predict the ports fishing vessels are heading for to deliver their catch, by utilizing the AIS data of the vessels. Obtaining accurate predictions will help the controllers make easier decisions on which port ports they should travel to, so that they can regulate the catch of the incoming vessels that they want, instead of having the risk of being located at the wrong port when a vessel they want to control comes in.

1.1 Problem Definition

If we are able to predict the port which a fishing vessel in heading to, the controllers are able to travel to that same port to control the catch on board of the vessel. The current solution to this problem is by manually collecting information from multiple different information channels. Hence, we want to investigate if it can be automated.

The thesis problem is therefore defined as the following:

> How can we use AIS data and machine learning to predict the destination ports of fishing vessels? We aim to investigate if combining AIS data with machine learning can make more accurate port predictions than a statistical baseline model.

1.2 Scope and Limitations

This thesis is based on publicly available data from different sources within the fishing industry in Norway, and thereby introduces some limitations, in addition to those we have applied to reduce the overall scope of this project.

- Norges Råfisklag covers the Norwegian cost north of Nordmøre, where most fishing vessels deliver their catch. Covering such an extensive Region of Interest (ROI) is beyond the scope of this thesis, as our aim is to assess the feasibility of our approach for predicting destination ports of vessels. Therefore, we reduce the ROI to the area illustrated in Figure 1.1,

restricting it to only include the county of Troms and Finnmark. Consequently, we focus exclusively on vessels heading towards ports within this region, and only consider AIS messages transmitted within this ROI. This specific area was chosen because fishing vessels often fish close to, or within, this region, which also contains a diverse range of ports.

- Because we use publicly available data, we are limited to only those fishing vessels included in this data, which are vessels with a minimum length of 15 meters.

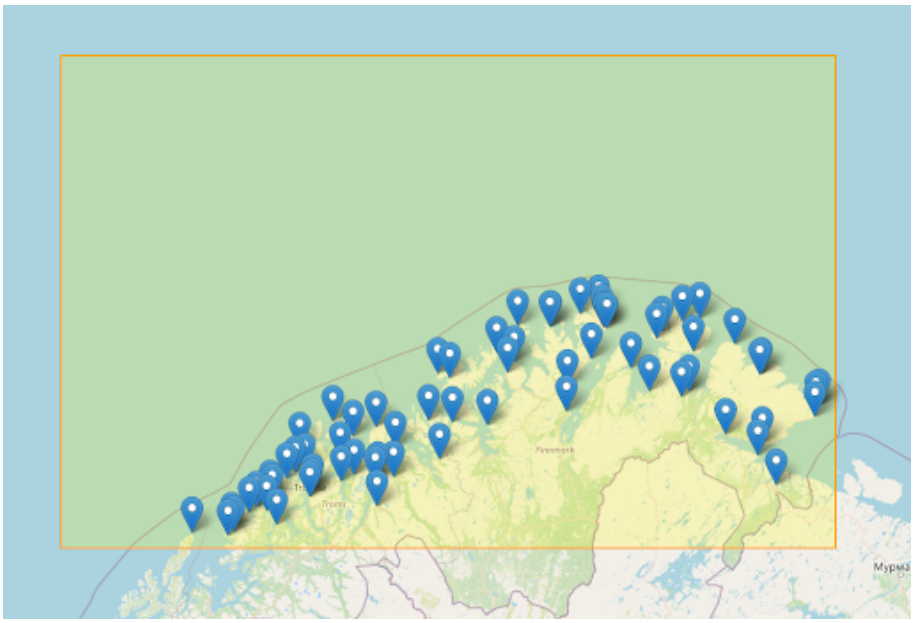


Figure 1.1: Region of interest with location of ports included

1.3 Contributions

This thesis makes the following contributions to solve the problem defined in the thesis statement:

- An extension to existing preprocessing methods, specifically aimed at fishing vessels.
- A simple statistical baseline model for predicting the destination port of a vessel.
- An algorithm for predicting the destination port of a vessel, based on the

predicted trajectories from a deep learning model.

- A tool for visualizing the predicted trajectories and destination ports.

1.4 Outline

- Chapter 2 describes relevant topics related to the work done in this thesis.
- Chapter 3 presents some related work to this thesis, and the field of predicting AIS trajectories.
- Chapter 4 describes the design choices we have made, and an overview of the preprocessing of the datasets.
- Chapter 5 goes into detail on how we implemented the different components in our work.
- Chapter 6 presents how we evaluated our approach, and the results from our benchmarks against our baseline model.
- Chapter 7 discusses relevant topics and alternative approaches, as well as future work.
- Chapter 8 concludes the work done in this thesis.

2 Background

This chapter describes the fundamental technologies and topics related to this thesis. Section 2.1 and Section 2.2 cover two messaging systems used by fishing vessels: AIS and ERS. Section 2.3 and Section 2.4 introduces the core libraries used throughout this project, and Section 2.5 gives an overview of the different file formats utilized.

2.1 Automatic Identification System (AIS)

Automatic Identification System (AIS) is an identification system used by vessels, which continuously transmits messages with information such as longitude, latitude, Speed Over Ground (SOG), Course Over Ground (COG), etc. to other AIS receivers within range. Such receivers can be other vessels, satellites, or dedicated AIS receivers at the shore.

The system was originally created as an anti-collision tool between vessels, but is today additionally used for monitoring vessel activity, which is useful for Search and Rescue (SAR), Vessel Traffic Service (VTS) to identify and monitor vessels at risk, or detecting illegal activities[1, 2]. The AIS system uses dedicated Very High Frequency (VHF) channels to transmit the messages, and has an average range of 40 nautical miles (but this can vary greatly depending on the conditions and type of AIS system being used).

AIS is not required for all vessels, but in the EU and Norway all vessels above 15 meters in length (inclusive) are required to use AIS. The messages are automatically sent by the system, and consists of three categories of information: static, dynamic, and voyage-related. Static information includes Maritime Mobile Service Identity (MMSI), International Maritime Organization (IMO) number, and radio call sign, which all are identification sequences for the vessel, as well as information about the size of the vessel. Dynamic information is longitude, latitude, SOG and COG, which we are using in this thesis. Voyage-information contains fields such as draft, estimated time of arrival, and destination, and is the only information in the AIS messages that can be altered. In practice, the quality of this information varies, and fields may be empty. It is generally not legal to turn off the AIS, however, there are certain exceptions when being "invisible" is permitted, such as when ships travel in areas where piracy occurs[3].

In this thesis, we use the AIS data to train a deep learning model to predict the future trajectory of fishing vessels. For each trip vessels have made within a time-period, we sort the AIS by MMSI to create tracks for each vessel, which form the training data of the model.

2.2 Electronic Reporting System (ERS)

In addition to fishing vessels using AIS, they are also required to use Electronic Reporting System (ERS). ERS is a system where the crew of the vessel has to manually send messages containing fishing-specific information, in contrary to AIS where the messages are automatically transmitted. The usage of this system is regulated by the ERS regulation[4], and covers topics such as what messages to send, which information is required in the message types, and when to send the messages. Fishing vessels above 10 meters in length are currently required to send ERS messages, but this requirement is set to be reduced from 10m to 8m in 2025[5].

There are many types of ERS messages, and most of the fields in an ERS message are repeated in the different messages types. These recurrent fields relate to the ID of the vessel and message, and timestamps. The most common ERS message types are Departure report (DEP), Detailed Catch and Activity report (DCA), and Port report (POR), and are the ones which are used in this thesis. The ERS datasets are publicly available[6] for vessels with a length of 15 meters or more, from 2011 and until today.

2.2.1 Departure report (DEP)

The DEP message, as shown an example of in Listing 2.1, is a message which is transmitted when the fishing vessel is departing from a port. This message contains information about at the estimated time the vessel departed from the port, the GPS position it is heading to, and the species code of the fish species it is aiming to catch. In this thesis, we utilize the *departureTime* field in the DEP messages to identify the latest possible timestamp a vessel is located at the destination port from its previous fishing trip.

```
{
    "id": 2,
    "messageVersion": 1,
    "messageType": "DEP",
    "radioCallSign": "RADIO87",
    "vesselMaster": "Tester",
    "transmissionTime": "2010-02-22T16:10:00.000+01:00",
    "storedTime": "2010-02-22T19:12:41.000+01:00",
    "modifiedTime": "2010-02-22T19:12:41.000+01:00",
    "recordNumber": 56,
    "group": "T",
    "registeredByFmc": false,
    "port": "NOAAF",
    "departureTime": "2010-02-22T16:10:00.000+01:00",
    "predictedStartTime":
       ↪ "2010-02-22T16:10:00.000+01:00",
    "predictedStartPosition": {
        "latitude": 0.75,
        "longitude": 34.0
    },
    "plannedSpecies": "ACH",
    "catchOnBoard": [
        {
            "speciesCode": "ACH",
            "quantity": 123,
            "unit": "KG"
        },
        {
            "speciesCode": "ALB",
            "quantity": 765,
            "unit": "KG"
        }
    ]
}
```

Listing 2.1: Example of ERS DEP message

2.2.2 Detailed Catch and Activity report (DCA)

Another ERS message type, and the most common one, is the DCA message. This message is sent between the DEP and POR messages sent by a fishing vessel, and contains information about the activity of the vessel. There are primarily two different types of activities a vessel conduct when sending the DCA message, steaming (STE) or fishing (FIS). When the *messageType* is STE, the vessel is sailing, and not fishing. This message usually only contains the identification and timestamps, which are the fields above the *departureTime* field in Listing 2.1. This message type also contains a *catchOnBoard* field if there is any fish on board, also seen in Listing 2.1.

When a DCA message has the *messageType* field set to FIS, it indicates that the vessel has finished its fishing activity. The message will then contain information about each of the hauls that have been made, each with information such as at what time the fishing started, the GPS position of the start and end of the haul, how long the fishing operation went on for, what species was caught, and the quantity of each species. This message can cover multiple hauls, as this DCA message is only required to be sent once every 24 hours.

We use the DCA messages of a vessel to be able to determine at what time the last haul of the trip occurred for a vessel, as after the last haul the vessel will head for a port. The time when the fishing ended is the earliest point where we start collecting AIS data of the vessel to construct its track until it has arrived at a port.

2.2.3 Port report (POR)

The POR message type, which an example of can be seen in Listing 2.2, is an ERS message that is sent when a fishing vessel is heading in towards a port. Some fields that are specific to this message type are *predictedTime*, which is an estimated timestamp when the vessel will be at the landsite of the port, and *transferQuantity*, which is the quantity of the catch onboard which is being delivered at the port specified in the message. The *port* field in the message is the international code of the port that the vessel is heading to, with *landsite* specifying the landsite, as a port can have multiple landsites within a given area. The *landsite* field is a text input field filled in by the crew of the vessel, which makes it prone to typing errors, or alternative names of the landsites.

We use the POR messages in this thesis to identify which vessels to retrieve AIS data for, based on their radio call sign identification (which we map to their MMSI number to pair with the AIS data). We also use the specified predicted time to get an estimate of when the vessel has arrived at the port, which

combined with the AIS tracks and next DEP message gives a better indication of when the AIS track should end.


```
{
    "id": 7,
    "messageVersion": 1,
    "messageType": "POR",
    "radioCallSign": "RADIO70",
    "externalRegistration": "T-1-H",
    "vesselName": "Test",
    "vesselMaster": "Tester",
    "transmissionTime": "2010-02-17T14:33:00.000+01:00",
    "storedTime": "2010-02-25T14:35:34.000+01:00",
    "modifiedTime": "2010-02-25T14:40:33.000+01:00",
    "rescindedTime": "2010-02-17T14:33:00.000+01:00",
    "recordNumber": 20,
    "group": "T",
    "registeredByFmc": false,
    "port": "NOLYN",
    "landsite": "kaia",
    "predictedTime": "2010-02-17T14:30:00.000+01:00",
    "quantityOnBoard": [
        {
            "speciesCode": "COD",
            "quantity": 270,
            "unit": "KG"
        },
        {
            "speciesCode": "HAD",
            "quantity": 64,
            "unit": "KG"
        }
    ],
    "transferQuantity": [
        {
            "speciesCode": "COD",
            "quantity": 270,
            "unit": "KG"
        },
        {
            "speciesCode": "HAD",
            "quantity": 64,
            "unit": "KG"
        }
    ]
}
```


Listing 2.2: Example of ERS POR message

2.3 Pandas

Pandas[7] is a powerful python data-analysis toolkit which was created in 2008. In 2009, it became open-source, and has since then been contributed to by the community, and is still being improved today. The library supports operations such as: reading and writing data between in-memory data structures and data formats (such as CSV, text files, etc.), dataset operations such as group by and merging, and slicing of large datasets.

The pandas dataframe is a data structure which supports these operations, and uses vectorization on the internal arrays used to create the dataframe to speed up the applied operations. Core algorithms of the pandas library are written i Cython[8] (C-extensions for python) to improve the speed when operating on data.

The pandas version used in this thesis is 2.2.2[9], and is used during the preprocessing of both the ERS- and AIS datasets.

2.4 Folium and Flask

Folium[10] is an open-source map visualization library for python, which builds on the mapping strengths of the Leaflet.js[11] JavaScript library. It allows for data to be manipulated using the python library, and then rendered as a Leaflet map in HTML. The development of Folium started in May 2013, and is still in active development today, with more than 150 contributors in total.

Flask is a lightweight web application framework for python which is designed to both set up simple web applications, but also have the ability to scale up to more complex applications. It started out as a wrapper around the Werkzeug[1] and Jinja [2] libraries, before today being one of the most used python libraries for web application development, used in 1.8 million projects[12].

In this thesis, we use Folium v0.16.0 to visualize our data, which together with Flask v3.0.3 form a simple web application where predicted trajectories and destination ports can be inspected in a map.

1. https://werkzeug.palletsprojects.com/en/3.0.x/
2. https://jinja.palletsprojects.com/en/3.1.x/

2.5 Data Formats

Data can be represented in multiple different ways based on how the data is going to be used. Some formats are easier to interpret, while others are more space efficient. This section will cover some of the data formats used in this thesis.

2.5.1 JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) is a lightweight data-interchange format which is easy for humans and computers to understand and create[13]. The data format is used to represent structured data, and often used when data is transmitted in web applications.

JSON stores its data in key-value pairs, where an arbitrary amount of these pairs construct an object. JSON can store multiple of these objects, with each object being independent of each other, meaning that objects can have different amounts and types of key-value pairs[14, 15]. This enables both complex and nested data to be stored within the JSON format.

Listing 2.3 shows an example of how some information contained with AIS messages can be displayed in JSON. The values can be retrieved by accessing the corresponding keys, e.g., to retrieve the MMSI number *123456789* the key *mmsi* is used.

```
[
    {
        "mmsi": 123456789,
        "timestamp": "2024-01-01T08:00:00",
        "latitude": 18.98760,
        "longitude": 69.67740,
        "sog": 13.5,
        "cog": 0.1,
    },
    {
        ...
    }
]
```

Listing 2.3: Example JSON data

This thesis uses the JSON data format when retrieving AIS tracks of fishing vessels. This data is then further used for training, validation, and testing of our models to predict the headed port for a fishing vessel.

2.5.2 Comma-separated values (CSV)

One of the downsides with the JSON data format is that it is not space efficient, which limits its use when large amounts of data are to be stored. An alternative format to store the data in such a case is to use Comma-Separated Values (CSV). The CSV format is often used when the data can be structured in a tabular and flat(non-nested) way.

Listing 2.4 shows an alternative representation of the data from Listing 2.3 using the CSV data format. Instead of having key-value pairs, each column represents the value of the "key" defined in the header (note that the header is not required, but useful for describing the contained data). Each column is separated by a separator, which in this particular example is a semicolon. The separator can however vary, but is commonly either a semicolon or a comma.

```
mmsi;timestamp;latitude;longitude;sog;cog
123456789;2024-01-01T08:00:00;18.98760;69.67740;13.5;0.1
...
```

Listing 2.4: Example CSV data

The raw ERS data that is collected for this thesis uses the CSV data format. We also store AIS messages in the CSV data format. The pandas[7] python library is used for reading and writing data in the CSV format, and we use dataframes from the pandas library to perform vectorized operations on the data.

2.5.3 Python Object Serialization

Pickle[16] is a python-specific object serialization module which implements binary protocols for serializing and deserializing python objects. The serialization process is often called "pickling", and the deserialization process "unpickling".

During serialization of a python object with Pickle, a set of rules are followed to convert the object into a pickle-compatible byte-stream. Any python object can be serialized, even custom ones, as the only requirement is that the object has to be reconstructable. A pickled python object consists of a sequence of opcodes[3] and followed by arguments. These byte-sequences are interpreted by a tiny virtual machine which perform memory operations to its stack and memory based on the opcodes from the pickled object. When the virtual machine

3. https://github.com/python/cpython/blob/3.12/Lib/pickle.py

reaches the *STOP* opcode, the original python object is located on the stack of the virtual machine, and returned as the deserialized python object[17].

The pickle format is used in this thesis to store the arrays for the training, validation and test sets, which contains fishing vessel trajectories constructed from their AIS messages.

3

Related Work

Predicting the behavior of vessels based on their AIS transmissions is a field which has been researched for many years, although not as much with port prediction in mind, especially for fishing vessels. This chapter presents some of the related work to this thesis.

3.1 AIS data-driven ship trajectory prediction modelling and analysis based on machine learning and deep learning methods

In 2023, the paper *AIS data-driven ship trajectory prediction modelling and analysis based on machine learning and deep learning methods* [18] was published. The aim of the paper was to analyze the performance of ship trajectory prediction methods to find their advantages and disadvantages in different prediction scenarios. The authors analyze and compare five machine learning methods, and seven deep learning methods, where the chosen methods were the most popular ones used in papers between 2000 and 2023 involving ship trajectory prediction[19]. The following five machine learning methods are analyzed in the paper:

- Kalman Filter (KF)

- Gaussian Process Regression (GPR)
- Support Vector Regression (SVR)
- Random Forest (RF)
- Back Propagation Network (BP)

The following seven deep learning methods were evaluated:

- Recurrent Neural Network (RNN)
- Long Short-Term Memory (LSTM)
- Bi-directional LSTM (Bi-LSTM)
- Gate Recurrent Unit (GRU)
- Bi-directional GRU (Bi-GRU)
- Sequence to Sequence (seq2seq)
- Transformer

The conducted tests were done using three separate AIS datasets at different locations along the coast of China: a port area, a complex traffic area, and an area around a promontory. In total, the datasets consist of 15 000 trajectories, and 17 000 000 data points (AIS messages). The size of the three datasets differ, and is categorized as small, medium, and large by the authors of the paper.

To evaluate the performance of the machine- and deep learning methods, six evaluation metrics were used:

- Mean Squared Error (MSE)
- Mean Absolute Error (MAE)
- Symmetrical Mean Absolute Percentage Error (SMAPE)
- Final Displacement Error (FDE)
- Fréchet Distance

- Average Euclidean Distance

The six evaluation methods were used for each model to measure the error rate of the predictions made by the model compared to the true trajectory of the vessel. The test-datasets contains 10% of the original trajectories. From the results presented in the paper, the best overall performing models are the Transformer, Bi-GRU and GRU model, with all of them being deep learning models. The worst-performing models is the BP neural network, followed by RNN and GPR. Overall, the deep learning methods perform better, with the difference in error rate increasing for complex trajectories.

The model performing the best on the different datasets varied depending on the size of the dataset (as well as the complexity of the trajectories in the datasets). The transformer model performed the 4th best on the small AIS dataset, best at the medium-sized, and 3rd best at the large-sized dataset. The accuracies of the predicted trajectories for the transformer model were 91%, 92%, and 91% respectively.

Since the transformer model is one of the most accurate methods for predicting trajectories, and the source code[1] of the tested model is publicly available on GitHub, we utilize the transformer model in this thesis to forecast the trajectories of fishing vessels.

3.2 TrAISformer – A Transformer Network with Sparse Augmented Data Representation and Cross Entropy Loss for AIS-based Vessel Trajectory Prediction

In the TrAISformer paper[20] the authors propose an approach using a generative transformer[21] architecture to forecast vessel trajectories based on AIS data. The aim of the work presented in the paper is to achieve accurate long-term predictions for vessel trajectories. In addition to presenting the transformer model, the paper introduces a new loss function used during training of the model.

TrAISformer is a modified transformer network, with the transformer part adapted from minGPT[2], designed to extract long-term temporal patterns from

1. `https://github.com/CIA-Oceanix/TrAISformer`
2. https://github.com/karpathy/minGPT

AIS trajectories. The model was originally trained on an AIS dataset provided by Danish Maritime Authority (DMA) and uses the new cross-entropy loss function during training. The dataset used to train and evaluate TrAISformer consists of messages transmitted by cargo and tanker vessels in Kattegat. Prior to preprocessing, their dataset comprised of roughly 710 million AIS messages in total.

The TrAISformer model utilizes the latitude, longitude, speed over ground, and course over ground as input features for each AIS message in the input trajectory. However, instead of the AIS input features being represented as a standard four-dimensional real-value vector, they are represented using *four-hot encoding*[22, 23], which consists of four concatenated one-hot encoded vectors (one for each AIS input feature). Each four-hot encoded vector is then paired with a high-dimensional embedding vector before being fed into the transformer network.

The results presented by the authors show a significant improvement over existing machine learning and deep learning models, both for short-term and long-term prediction. The prediction error is calculated by measuring the haversine distance between the predicted and true locations of the vessel. The mean prediction error of the model is reported as 0.48, 0.94, and 1.64 nautical miles for 1, 2, and 3-hour ahead predictions. This error is calculated using the best-of-N principle, where the best predicted trajectory out of $N = 16$ is used in the calculation. The input length during this evaluation is three hours, based on our re-run of the benchmark of the TrAISformer model.

In this thesis, we train the TrAISformer model on our own dataset, consisting of AIS messages from a custom ROI. The AIS messages we use belong to fishing vessels heading towards ports located in Troms and Finnmark, Norway, to deliver their catch. To predict the destination port of a vessel, we use the TrAISformer model to predict trajectories for the vessel as the first step. These trajectories are then used as input for our port prediction algorithm, which provides a probabilistic prediction on which port the fishing vessel is heading towards to deliver its catch. Both the source code[3] of the TrAISformer model, and the code they used for preprocessing[4] AIS data, are publicly available on GitHub.

3. https://github.com/CIA-Oceanix/TrAISformer
4. https://github.com/CIA-Oceanix/GeoTrackNet

3.3 Trajectory pattern extraction and anomaly detection for maritime vessels

This paper[24] from 2021 focuses on extracting trajectory patterns for maritime vessels. The authors analyze these patterns using methods presented in the paper to make various predictions. The first group of trajectory analysis predicts the arrival port, arrival time, and trajectory of a vessel based on its AIS messages. The second type of trajectory analysis detects anomalous behavior of maritime vessels, also based on AIS messages.

In regard to port prediction, multiple conventional classifiers are used, such as decision tree, random forest, and multilayer-perceptron. A LSTM model adopted from [25] is also tested. The AIS input features include latitude, longitude, speed over ground, and course over ground, similar to the TrAISformer model. The arrival port label is represented using one-hot encoding.

To enhance the accuracy of the classifiers, clustering is used, with DBSCAN[26] performing the best among the tested clustering algorithms. Ten clusters are created, and for each cluster, a version of the classifier is trained. This results in ten differently-trained versions of the classifier, each responsible for making predictions within its geographical region.

The results are measured using 10-fold cross-validation. The random forest classifier performed the best among the tested classifiers, achieving a port prediction accuracy of 86%. However, its per-port accuracy reveals that there is a significant difference in accuracy based on which port the AIS trajectory is heading towards, with accuracies as low as 8% and 11% for certain ports. The LSTM model had an overall accuracy of 65%, but its accuracies were the most consistent across all ports.

The authors of this paper aim to solve a similar challenge to this thesis: predicting destination ports based on AIS trajectories, although not for fishing vessels. The AIS input features used in their models are the same as in this thesis, achieving accuracies of 86% and 65%. The main difference between their work and this thesis is the deep learning models used for the AIS trajectory predictions. According to Section 3.1, the transformer model achieves a higher accuracy in AIS trajectory predictions. Higher accuracy trajectory predictions should lead to a greater port prediction accuracy, due to the trajectories being more accurate. However, the results are not directly comparable, as ours and their number of ports varies, 68 vs. 8, and their ports are relatively spaced out over a region which is of equal size, or slightly larger, than ours.

4

Design

This chapter describes the design and design choices made in regard to predict the destination port of fishing vessels. In Section 4.1 the behavior of fishing vessels are described, and how we can utilize AIS and ERS to create a dataset of vessel trajectories heading for a port. Section 4.2 gives an overview of the preprocessing steps to create this dataset. Section 4.3 and Section 4.4 covers the design of two unique models aiming to predict the destination port of vessels, which in Chapter 6 are benchmarked and compared against each other.

4.1 Extracting the Vessel Trajectories

This section describes key observations in regard to the trajectories of fishing vessels, and how we combine different messaging systems used by fishing vessels to extract the part of the trajectory where the vessel is heading towards a port.

4.1.1 The Nature of Trajectories for Different Vessels

Previous work on AIS trajectories of vessels have been working on cargo vessels, or similar vessel types which deliver goods from location A to location B. These voyages have a specific destination in mind, and a pre-planned route where the aim is to reach the destination port in the most efficient way possible. The

trajectories of such voyages will therefore be more or less straight-lined when possible, with minimal deviation from the planned route (unless something unexpected happens).

Trajectories of fishing vessels however, as illustrated in Figure 4.1, does not behave in the same way as other ships. In the figure, the red lines indicate that the fishing vessel were only sailing (also called *steaming*), while the blue lines indicate that the vessel were conducting fishing activity. Comparing the trajectories of the fishing vessel while fishing and cargo vessels, there are no similarities in the trajectories, as the trajectory of the fishing vessel perform seemingly random movements. Comparing the trajectory of the fishing vessel while steaming to a cargo vessel's trajectory however, there is a resemblance of similarity in the parts of the trajectory of the fishing vessel when it departs from the port, and when it is heading towards it. This is because after the last fishing haul is over, the vessel has a destination port in mind, and similarly to other vessel types, want to head in towards the port efficiently.

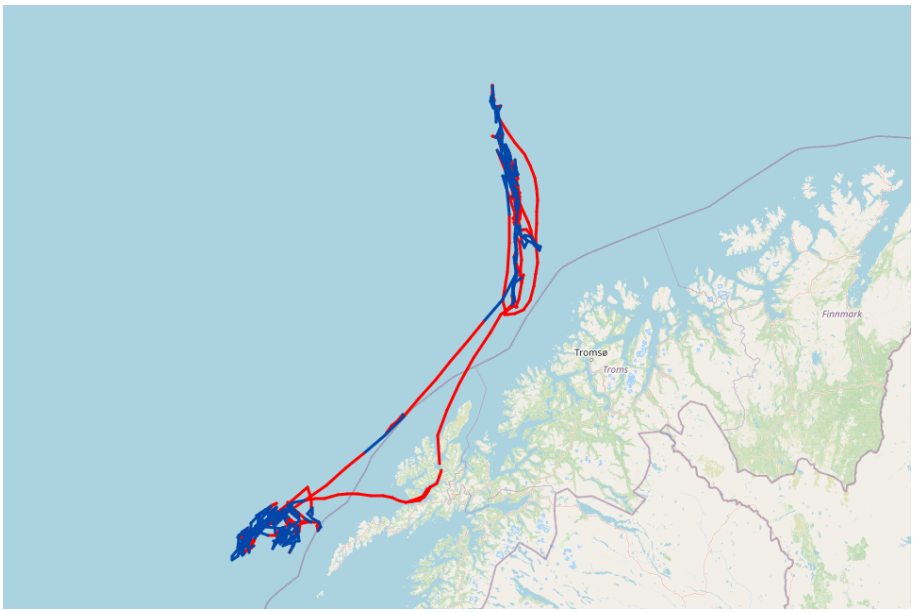


Figure 4.1: Trajectory example of a fishing trip

Using these observations, it is possible to conclude that training any model on the whole trajectory of a fishing vessel will most likely not be effective, because of the random nature of the trajectories when the vessels are fishing. However, the part of the trajectory which is after the last fishing haul of the trip is possible to train a model on, as at this time the vessel will mostly likely already know which destination port it is heading for. The trajectory of the vessel at this stage will then be similar to any other vessel types with a destination in mind: streamlined. Because of this, we only want to use the part of a fishing vessel's

trajectory after the last fishing haul is over when determining its destination port.

4.1.2 Combining Vessel Reporting Systems

There are two main challenges when it comes to define where in a trajectory a fishing vessel is starting to head towards a port: finding where the last fishing haul ended, and determining when the vessel has reached the port. Multiple papers have been written about the subject of pinpointing where a vessel has been fishing, based on only its AIS trajectory [27, 28, 29]. Their solutions are based on utilizing the abnormal patterns arising during fishing, compared to when the vessels are only steaming, which also can be seen in Figure 4.1.

To determine when the vessel has reached its destination port, it is possible to look at the speed, and the duration spent within a region, to identify when the vessel is anchored. If the locations of the possible destination ports are available, the distance between the port and the vessel can be measured as well, to identify when it has reached the port.

While these both are viable options to solve the challenges when extracting the *correct* part of a fishing vessel's trajectory, the ERS messages transmitted by fishing vessels can be used also to solve both of these challenges. The last DCA message transmitted by the vessel before a POR message reveals when the last fishing haul ended. Therefore, the time of this message can be used as an indicator for when the fishing haul ended (because the message is sent after the haul is over). It would also be possible to use the coordinates of where the haul ended as an alternative method, but these are not necessarily precise enough to combine with the AIS messages without applying some accepted error margin between the ERS coordinates and the AIS coordinates.

To determine where the trajectory stops, i.e., that the destination port has been reached, a combination of the ERS message types POR and DEP are used. The POR message gives an indication for when the vessel will reach its destination port, but is not accurate enough in its own to be used as the timestamp for when the AIS trajectory stops. Therefore, we also use the succeeding DEP message, which indicates the next departure of the vessel, because we know with certainty that in the time interval between the given arrival estimate, and the start of the vessel's next trip, the vessel will have reached its destination port. Using Equation 4.1, the mid-timestamp is calculated between the estimated arrival time and the timestamp of the next departure.

$$\text{mid-timestamp} = \frac{\text{<POR estimated timestamp>} + \text{<DEP timestamp>}}{2} \tag{4.1}$$

Because there is a high likelihood that the vessel will be at the port at this given time, the most recent AIS message sent before this timestamp is defined as the end of the trajectory. This approach of finding the end of the trajectory gives a high likelihood that the vessel will be at its destination port, but it is not guaranteed in the case where the estimated time is wrong by a large margin. Potentially, if the estimated time is a long time before the arrival, or a long time after the time of the next departure, the vessel can be outside the area of the port. While this were not an issue in the dataset we used, additional measures were added to ensure that the end of a trajectory was inside a five-kilometer radius of a port. This geographical check also removes the fishing vessels which steamed through our ROI on its way to another destination port.

4.2 Dataset Preprocessing

Because this thesis works with raw data which originates from different systems, preprocessing is required to extract and merge the data such that only the vital information is left. The following section gives an overview of the preprocessing of this data.

4.2.1 ERS Dataset

The raw ERS dataset is retrieved from the Directorate of Fisheries[6], and contains the ERS messages transmitted by vessels larger than 15 meters within the Exclusive Economic Zone (EEZ) of Norway. The dataset does not contain all the possible ERS message types, only DEP, DCA, POR, and TRA. In this dataset there are 4 500 000 ERS messages transmitted between January 1st 2016 and December 31st 2023. Using the definition of a fishing trip from Section 4.1, where a fishing trip is required to contain at least a DCA (with fishing activity conducted, not only steaming), POR and DEP message, the ERS dataset contains to 170 000 fishing trips within Norway's EEZ, before being combined with the AIS messages.

4.2.2 AIS Dataset

The AIS dataset, consisting of 745 000 000 messages, are fetched from the Application Programming Interface (API) of Kystdatahuset[1]. All AIS messages transmitted within our defined ROI, within the timeframe January 1st 2016 to December 31st 2023, are collected where the speed of the vessels were at least 0.5 knots. Messages where the vessels have a speed less than 0.5 knots are not retrieved, because they are likely either moored, or close to docking at a port. Including such messages in the dataset would increase its size significantly, as the close-to-shore transmission rate can be as low as a couple of seconds between each message, and does not change or impact the overall trajectory of the vessel. It only increases the number of messages that has to be removed at a later data-processing stage. Subsection 4.2.4 describes the process of how the AIS tracks are constructed when combined with the ERS messages of the vessel.

4.2.3 Destination Ports Dataset

In Norway, there are 244 different facilities[2] within the geographical region that Norges Råfisklag covers where fishing vessels can deliver their catch. Most of these locations have ports where the vessels dock, but some are fish farms or shops. Our port dataset consist of facilities within our defined ROI, but has also been manually reviewed down to 68 unique ports where vessels can deliver their catch. Ports which are close to each other are downsampled to a single port, because they either share the dock, or are so close to each other that a controller is able to move the short distance to the other port, without losing the possibility of controlling the catch of the vessel. This downsampling is only done for ports where there is no practical significance for the controllers.

4.2.4 Assembling the Tracks

For each of the 170 000 fishing trips defined by the ERS messages, the vessel's trajectory heading towards its destination port is reconstructed. AIS tracks are assembled by collecting the AIS messages transmitted between the last fishing haul of the vessel and its arrival at the port. However, because the ERS dataset consists of messages transmitted within Norway's EEZ, and not exclusively our ROI, additional preprocessing steps are required. These preprocessing steps ensure that the finalized dataset is of sufficient quality. These steps are described in detail in Chapter 5.

1. https://kystdatahuset.no/
2. https://rafisklaget.no/mottakskartet

After preprocessing, the dataset consists of 2 000 000 AIS messages, distributed across 17 000 unique tracks. These tracks are randomly distributed into three subsets: training, validation and test sets, containing 80%, 10%, and 10% of the total tracks, respectively. There are no duplicate tracks across these subsets. The training and test sets are used for training and evaluating both the baseline model and the transformer model, while the validation set is only used during the training stage of the transformer model.

4.3 Baseline Model

In order to evaluate the performance and effectiveness of using a deep learning approach to predict the ports fishing vessels are heading for, a simple statistical baseline model has been developed to compare against. The baseline model is lightweight, and developed with simplicity in mind. The goal of this model is to test if complex models are required to accurately being able to predict the port which fishing vessels are heading for.

The architecture of the baseline model is illustrated in Figure 4.2. The model uses a mapping between the MMSI number and the ports which the vessel has headed to previously. A probability is attached to each port to indicate the probability of it being headed to by the vessel. The probabilities are determined in the training phase of the model. The training and evaluation of this model uses the same dataset as the deep learning model to assure fairness in the comparison of the two models.

4.3.1 Training the Model

The training process of the baseline model can be divided into two stages. The first stage in the training of the model is responsible for creating the initial mapping between the MMSIs and the ports, i.e., setting up the architecture shown in Figure 4.2. For each AIS track, the MMSI and destination port are extracted and put into their respective mapping location. At the end of this training stage, each MMSI entry in the model will store the destination ports of the vessel, and the number of times the port was visited by it.

In the second stage of the training, the probabilities for each MMSI visiting each of the destination ports are calculated. The probability for a single MMSI number of the vessel visiting a port is given in Equation 4.2. The formula calculates the probability by looking at how many times a port was visited out of the total number of visits. The sum of all observed port-visits for a single vessel is equal to 1.

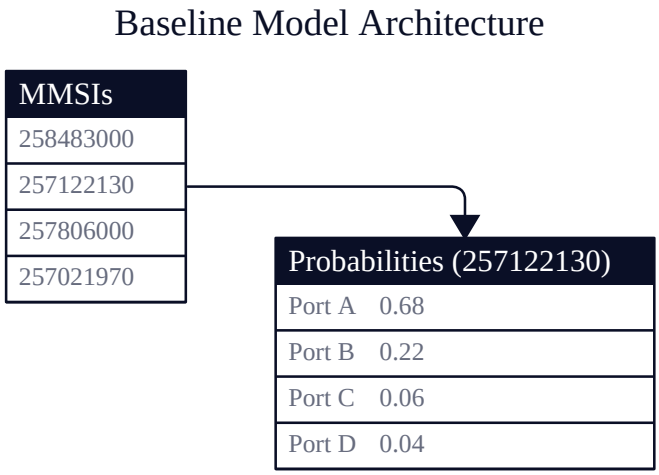


Figure 4.2: Baseline model architecture

$$\text{port\_probability} = \frac{\text{n\_port\_visits}}{\text{n\_total\_visits}} \tag{4.2}$$

When each port has been given a value between 0 and 1, indicating the probability of each port for the given MMSI, the ports are ordered by their probability in descending order.

4.3.2 Predicting the Destination Port

Because of how the baseline model is set up, retrieving the most probable destination port for a vessel is efficient and fast. A MMSI number is required to make a prediction. The MMSI number is used as the lookup key by the model to access the destination ports and their corresponding probabilities. As the training phase orders the ports in descending order based on the frequency in which they have been visited, the port that the fishing vessel is most likely to have as its destination is located at index 0. The second most probable port is at index 1, etc. In cases where the MMSI does not exist in the training set, the model is not able to make a prediction on which port is most likely to be the

vessel's destination port. When the MMSI exists, the highest probability port is predicted as the destination port of the vessel, regardless of the location of the vessel.

4.4 Deep Learning Model

This approach's process of predicting the destination port of a vessel given its recently transmitted AIS messages is divided into two main stages: predicting the future trajectory of the vessel, and decide which the predicted trajectory is heading towards. For the first stage, the TrAISformer[20] deep learning model is used. For the second stage, the main method to decide which port a vessel is heading for is by measuring the distance to the possible destination ports. However, due to challenges arising for using an approach only measuring the distance to the ports, additional design measures are added to improve the accuracy of the port predictions.

4.4.1 Trajectory Prediction using TrAISformer

While the underlying structure of how the TrAISformer model remains unchanged from what is presented in the original paper[20], the configuration of the model has been changed slightly to suit the dataset used for training and evaluation. One of such changes is the minimum speed requirement, which is changed from 1.5 knot to 0.5 knots. The speed requirement were required for the model to not cut the training tracks too far away from the shore/ports, as fishing vessels will travel with a speed of less than 1.5 knots when they get closer to their desired port. These close-to-port messages are something we want when we are going to decide which port the vessel is heading for. Because of this same reason, the removal of AIS messages closer than one nautical mile to the shore is not applied, as our goal is to find the destination port of the vessel, and not only the future trajectory of it.

Originally, the predictions of the TrAISformer code were only given in a context of evaluation of accuracy. However, since we need these for port prediction purposes, a change was necessary so that the trajectories could be retrieved from the model. This change is necessary to make it possible to create predictions for single tracks which in a production environment would arise from outside the dataset the model uses for training, validation, or testing.

When the model makes a prediction for a given input track, it produces N predicted tracks as output. These tracks are unique due to temperature being incorporated into the model, which adds some randomness to the predictions.

N is an adjustable parameter, which is set to 16 in the original paper. This thesis uses the same $N = 16$ for the number of predictions made for a single input track, although any other number could be used, preferably > 10 so that it is possible to make probabilities for the possible destination ports in the port prediction step.

Figure 4.3 and Figure 4.4 show $N = 10$ predictions made for a single input track. The track given as input is colored as a white line, and consists of 12 AIS messages in both figures (corresponding to one hour of movement for the vessel). The red colored track which ends at a port in both figures is the actual trajectory the vessel took when heading to the port. The other-colored lines are the N predictions made by the TrAISformer model given the white input. This color-coding consists throughout the figures where trajectory predictions are made.

Figure 4.3: Example where the look-ahead time is not too far off

For both Figure 4.3 and Figure 4.4 the look-ahead time, i.e., how far ahead in time the model predicts the trajectory of the vessel, is set to 2.5 hours (corresponding to 30 data points). Both the input and output data to/from the model is given as a time series, but visualized as lines for easier interpretability of the trajectories. Figure 4.5 shows the actual output from the model, with the raw output (figure a), and with smoothening applied (figure b). The smoothening is applied, using a moving average, for all predicted output tracks to make them more similar to actual vessel trajectories. Subsection 5.3.2 goes into further details about the moving average algorithm.

The figures (Figure 4.3 and Figure 4.4) highlight one of the main issues when it comes to deciding how far into the future the model should predict the

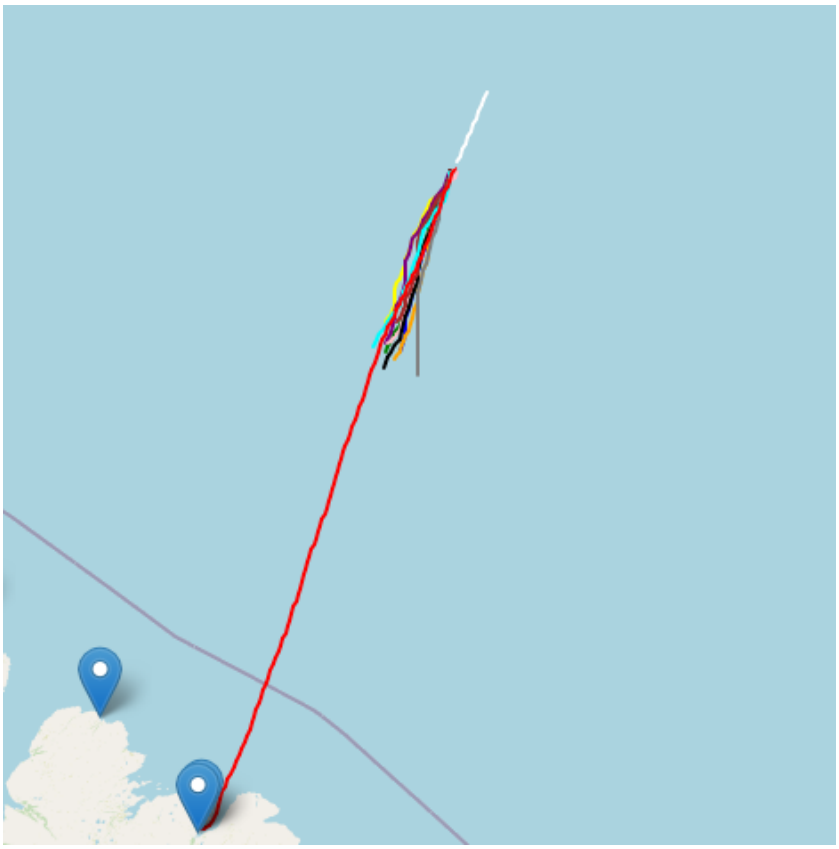
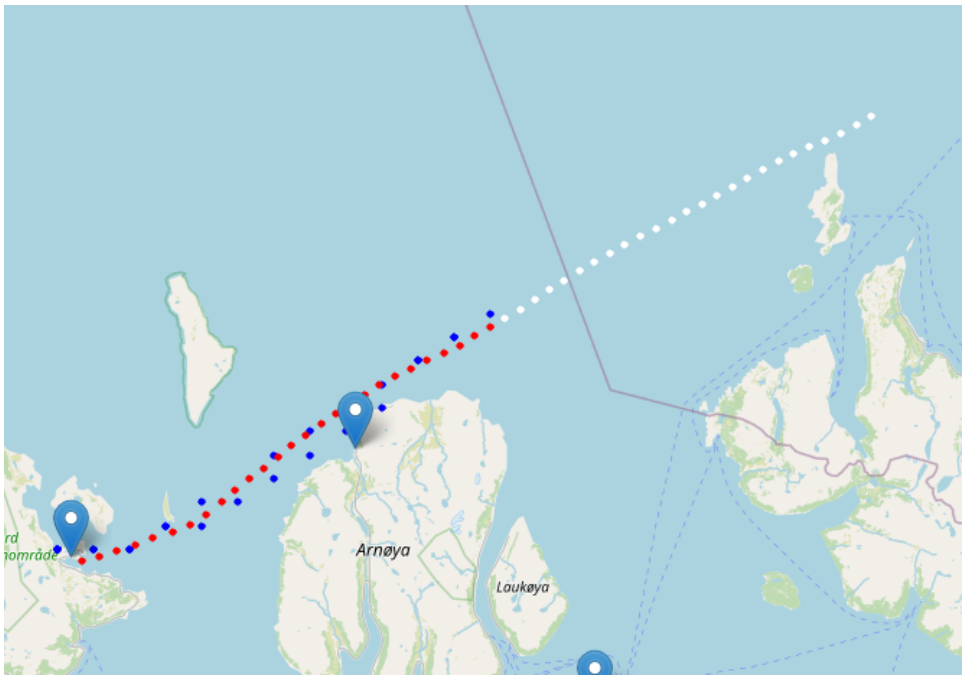


Figure 4.4: Example where the look-ahead time is very far off

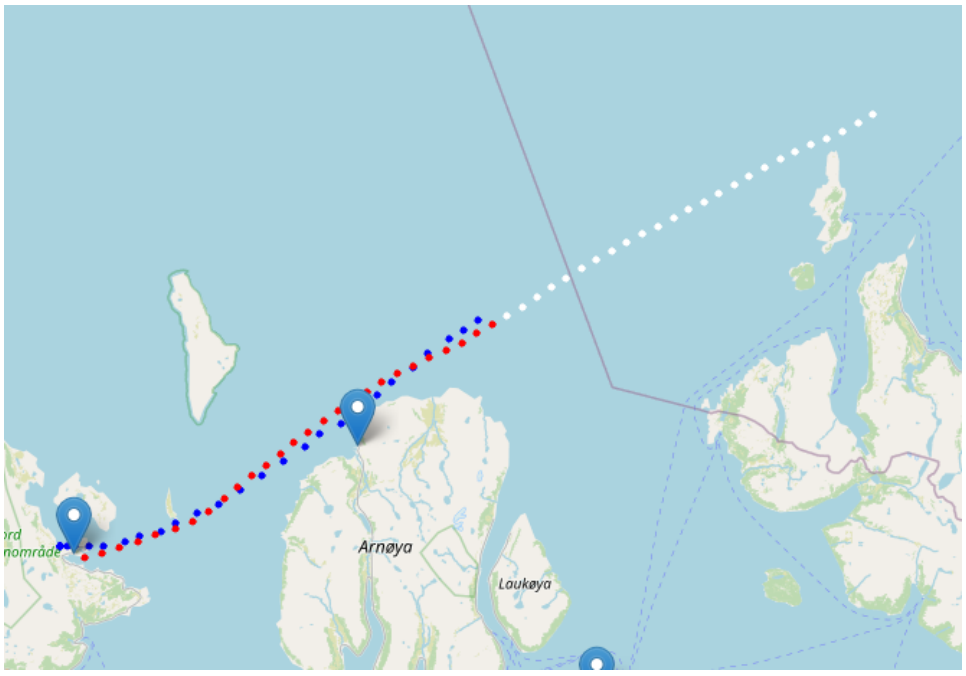
trajectory of the vessel. While the look-ahead time for the predictions in Figure 4.3 end not too far from a port (the closer the better), the same look-ahead time for the predictions in Figure 4.4 show that there is no universal look-ahead time which can be used to make the TrAISformer model predict trajectories which end at a port.

Two possible solutions were considered to the problem of deciding how far into the future the model should predict the vessel trajectories. The first solution involves incrementally increasing the look-ahead time of the model with some step size (e.g., 10 minutes). The advantage of such method is that the predicted trajectory will never become too long, as at some time the trajectory will be close to a port. This removes the problem of some tracks requiring a farther look-ahead time than others to reach a port.

The disadvantage of such an approach is the time and computing resources required to make a prediction for a single vessel where the vessel is far from a port. This method could also have unwanted behavior when the predicted



(a) Without moving average applied



(b) With moving average applied

Figure 4.5: Single-Prediction output (blue dots) from TrAISformer model, visualized with/without moving average

trajectory is close to a port, similar to the example in Figure 4.3, as it could seem like the vessel is heading towards the port being passed by. The issue arises when deciding if a port that is close to the trajectory is the destination port, or if in some arbitrary time in the future the trajectory will be close to the actual destination port. Additionally, this approach would be slow if the vessel is far from the destination port, as for each time step predicted an algorithm would have to check if the trajectory is at the destination port.

The second solution makes the model predict far ahead immediately, such that the predicted trajectory likely at some point will have headed to the destination port. If the vessel is close to shore at the time of prediction, the predicted trajectory will likely head to the port, and eventually past it. The advantage of this solution is that only a single far-ahead prediction is required to be made, and the port prediction algorithm then only have to find where in this trajectory the vessel behaved as if it were anchoring at a port. This would solve the main issue from the first solution, as at any point in the trajectory where the predicted position of the vessel is close to a port, the future path of the trajectory would still be known (unless at the end of the predicted trajectory, but then the vessel is likely too far from shore for make an accurate prediction).

The disadvantage with this solution is that the model will in most cases predict parts of the trajectory past the destination port, often ending up on land, or close to another port by random chance. An example of such behavior is shown in Figure 4.6. In this example, and similar cases which can occur if the model attempts to predict farther ahead in time than the time it takes for the vessel to reach the port, it shows that the model is not trained on how the trajectory of the vessel should behave. Therefore, the next predictions spread, have abnormal geometric shapes, or even head on land. This issue however can be omitted with extra steps in the port prediction algorithm.

Because of the cost in the first solution to incrementally increase the length of the N predicted tracks, checking for each iteration if the predicted tracks are at a port, the second solution was chosen as the strategy when predicting the trajectories of a vessel. The model is set to predict ~8.5 hours ahead for all predictions it makes, independently of the current location of the vessel and distance to any port. This time corresponds to 100 data points of time series output from the model, and during testing resulted in the port prediction algorithm to locate at least one port for each vessel in more than 90% of the test cases, with an input track length of 30 minutes. This means that for the remaining cases, the coordinates of the input track were most likely more than 8.5 hours from shore. This percentage reduces when more input data is collected, as the vessels get closer than 8.5 hours to shore.

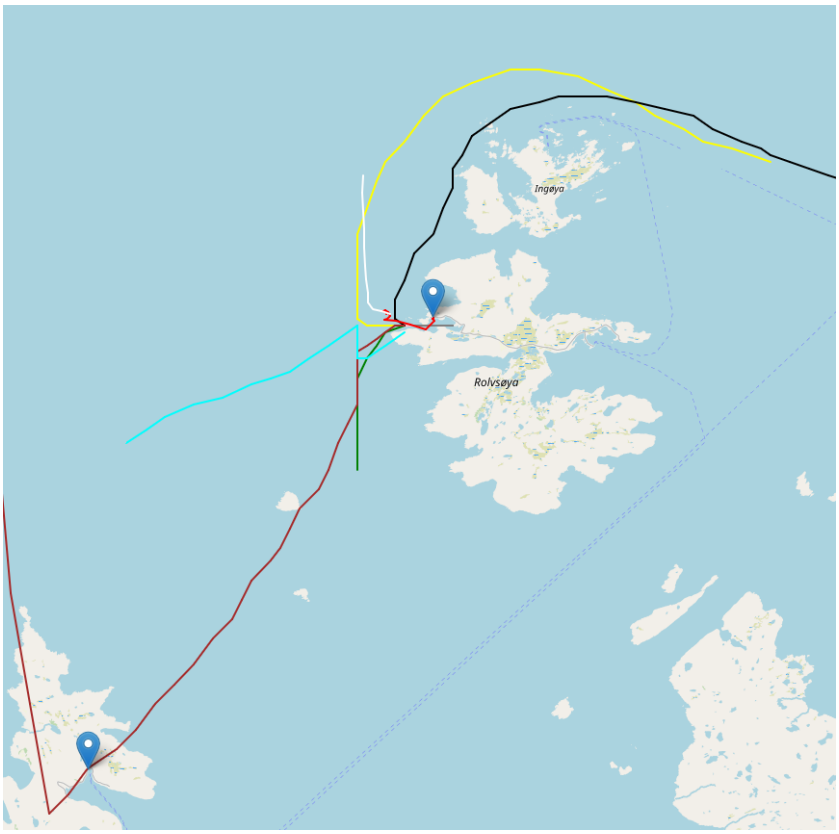


Figure 4.6: Predicting the trajectory past when the vessel reaching the port

It is possible to make predictions further ahead than 8.5 hours, but increasing this time frame also increases the inaccuracy of the predicted trajectories. Additionally, vessels this far out in the sea might not have finished their fishing activities. From the standpoint of a controller, knowing the destination port of a vessel 6, 8, or 10 hours in advance does not necessarily make a difference, as this is sufficient time to move to the port if it is near the area where they are performing controls. It should however be noted that there is nothing wrong with increasing or decreasing this value.

4.4.2 Destination Port Prediction

The TrAISformer model outputs 16 different predicted trajectories for a single vessel, each being time series consisting of data points with the attributes: latitude, longitude, speed over ground, and course over ground. For each trajectory, the steps shown in Figure 4.7 are taken in order to decide which port the predicted trajectories are heading for. Because each trajectory is handled independently, the number of unique ports all the trajectories are heading for

may vary between 0 and 16.

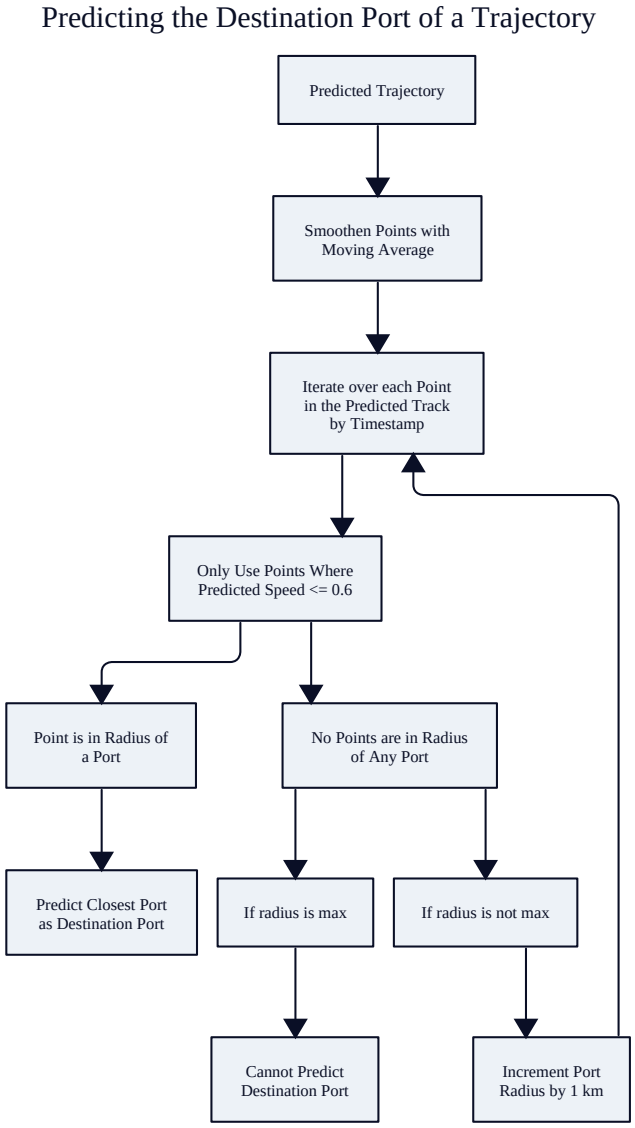


Figure 4.7: Diagram of the program flow to predict a destination port for a track

One of the main challenges the port prediction algorithm aims to solve is how to differentiate if a vessel is heading for a specific port, or just passing by it. An important difference in these two scenarios is the speed of the vessel. A

vessel approaching a port will reduce its speed when it is getting closer to its destination port, while a vessel passing by a port will maintain its current speed. Using Figure 4.8 as an example, the blue trajectory will have its speed lower the closer it is to the port it is heading for, while the black trajectory will maintain its speed, because it is heading for another port.

Figure 4.8: Predicted trajectories heading towards different ports

The difference in speed when a vessel is approaching a port, contrary to passing by it, is reflected in the dataset the TrAISformer is trained on. Therefore, the model is not only useful for predicting vessel trajectories, but also its speed (and course, as this is another attribute in the training data). The lowest speed occurring in the training dataset is 0.5 knots, and this is the approximate speed the model will predict for a vessel when it is closing in on a port. Therefore, the first step in the port prediction algorithm is to filter out the data points where the predicted speed of the vessel is more than 0.5 knots (we use 0.6 knots to have some leeway, as the predicted speed is not always exactly 0.5 knots when closing in on a port).

The remaining data points in the predicted trajectory will be those where the vessel has a predicted speed <= 0.6 knots. These points can come from different parts of the predicted trajectory, so one must decide which point(s) belong to the vessel approaching its destination port, and not because of other reasons where it may lower its speed. If the vessel is too far from shore when its trajectory is being predicted, the predicted trajectory might also not reach any port at all.

A simple solution to choose the destination port would be to choose the closest port at the first occurrence where the speed is less than 0.6 knots, but this could be an inaccurate prediction if the transformer model predicts the vessel will slow down when being very far from shore, or if it slows down for other reasons, without being at the destination port.

Instead, we define a radius for each of the ports, so that each port is the center of a circle-shaped area. With this, a port can only be considered as a destination port if a data point from the vessel's predicted trajectory is located inside this circle. However, one issue with this approach is to decide the size of the radius for each port. The predicted trajectories do not stop at the exact location of a port, but often in the surrounding area, so the radius should not be too small, or else the trajectory will miss it. The radius should also not be too large, as this can cause overlapping areas between different ports, making it harder to decide which one the trajectory is heading towards.

Because of the issue with the size of the radius, the port prediction algorithm starts with a small radius for each port (because if a trajectory is within a small radius, it is more likely to have headed for that specific port). If no parts of the predicted trajectory are within any port's area at any point in time, the radius is increased for all the ports. The algorithm is then repeated, checking if any parts of the trajectory are within the now larger radii of any ports.

The increasing of the size of the areas surrounding the ports continue until a maximum radius has been reached for the areas. If there still are no parts of the predicted trajectory located within the area of any ports, the algorithm considers the uncertainty to be too large to predict a destination port for this single predicted trajectory (because each trajectory is handled independently, and the TrAISformer model outputs 16 trajectories for each input). The algorithm not predicting a destination port can occur if the predicted trajectory is inaccurate, i.e., not heading close to any ports, or if the vessel is more than 8.5 hours from shore. The reason for both these cases are likely that the input to the TrAISformer model is too small, and can be solved by retrieving more AIS messages from the vessel.

When a destination port, or no port, has been predicted for each of the 16 trajectories predicted by the TrAISformer model, the probability of the vessel visiting each port is calculated. The probability of a port being the vessel's actual destination port increases based on how many of the 16 trajectories were predicted for that particular port. Ports where none of the trajectories were predicted to visit are given a probability of 0.

5 Implementation

This chapter describes the implementation details of important components of this thesis. The preprocessing and construction of AIS tracks is described in Section 5.1 and Section 5.2. In Section 5.3 the implementation details of the destination port prediction algorithm is covered. Section 5.4 describes the inner workings of the web application used for visualizing predicted trajectories and ports in an interactive map.

5.1 Dataset Preprocessing

This section goes into further details about the initial preprocessing of the raw ERS and AIS data, before the construction of AIS tracks.

5.1.1 Preprocessing Raw ERS Data

The DCA, POR and DEP ERS messages originate from different CSV files when downloaded from the Directorate of Fisheries[6]. Dataframes from the pandas[7] python library are used to store and perform operations on the raw ERS data.

Figure 5.1 illustrates the process of cleaning the ERS datasets to only contain relevant fields for the purpose of defining the start- and stop time from when

a vessel is done fishing until it has reached a port. For each ERS message type, which unprocessed are originally stored in separate CSV files, the relevant fields are extracted. The ID of a message, as well as the radio call sign for the vessel transmitting the message, are common fields for all the ERS message types. Each ERS message type additionally have a unique field which is extracted. For POR messages, this is the field giving the estimated time of arrival at its destination port. For the DEP messages, this unique field is the departure time of the vessel, i.e., the timestamp when the vessel left the port.

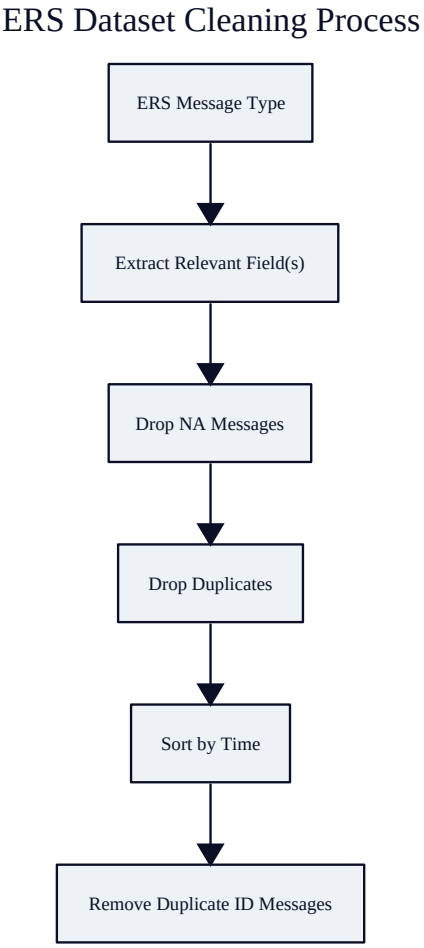


Figure 5.1: Diagram of the process of cleaning the ERS dataset

In the DCA messages, there are two important fields, which describe the timestamp where the fishing haul started, and the duration of the fishing haul. Using these two fields, the time when the fishing haul ended can be deduced, and is extracted from this message type. However, not all the DCA messages contain this field, because the DCA message type is split into two categories of activity, STE (steaming) and FIS (fishing). STE messages do not contain any fishing haul information, and is therefore filtered out before deducing the stop time of the fishing haul.

Duplicate messages, or messages containing NA entries in the extracted fields, are removed from the ERS datasets. Messages with duplicate IDs are reduced to only a single message. This occurs especially for the DCA messages, as if multiple fishing hauls are included in the same DCA message there are multiple equal-ID entries in the CSV files, because the CSV file format does not support nesting of data. DCA messages are only required to be sent once every 24 hours, so information about multiple hauls is often included in these messages.

As there are no fields in the ERS messages which can be used to identify which ones have been transmitted during the same trip (across DCA-, DEP-, and POR messages), the extracted time fields, as well as the radio call signs are used. A left outer join operation is performed with the POR and DCA messages, with the radio call sign being the field which has to match for two messages to be considered being joined together, using the pandas built-in *merge_asof*[1] function. It will attempt to match each POR message with a DCA message, based on the fishing haul stop time (in the DCA messages) and the estimated time of arrival (in the POR messages). The DCA message which has its haul stop time the closest, and earlier than, the ETA in the POR message gets matched with it. Because the left outer join operation includes POR messages which have not been matched with any DCA messages in the output, these non-matched entries are removed afterward. The result from this operation contains pairs of POR and DCA messages, where the DCA fishing occurred first, followed by the POR.

The outer left join operation is also performed between the result of the previous join operation (POR and DCA) and the dataset containing the DEP messages. However, because the DEP messages are intended to not be for the same trip (i.e., the start of the trip the DCA and POR occurred on), but rather the *next* trip, the join operation searches forward for the DEP message sent after the POR by the vessel. Entries which did not have any match were also removed after this join. The resulting dataset from these join operations yield entries with information from a DCA-, POR-, and DEP message, where the events (fishing,

1. `https://pandas.pydata.org/docs/reference/api/pandas.merge_asof.html`

arriving at port, leaving port) corresponding to those message types occurred in that sequential order for the vessel.

5.1.2 Preprocessing Raw AIS Data

The API where the AIS data is fetched from allows for a polygonal region to be defined, such that only AIS messages transmitted within this region are included in the response. Some additional steps of filtering are also applied to ensure better quality of the data, as shown in Figure 5.2. Messages with NA fields, and duplicates, are both removed from the dataset. The SOG of the vessel should be within the speed range 0.5-30 knots to not be filtered out, and the COG of the vessel must be within 0 and 360 degrees. This filtering ensures that messages with unnaturally high or low values are included in the dataset (in regard to the lower speed limit, it is set to remove messages where the vessel is almost standstill). The messages that pass these stages of filtering are stored in CSV files for further usage, which is when they are matched with the ERS messages to assemble the AIS tracks of the vessel heading to a port.

5.2 Constructing AIS Tracks

The preprocessing steps taken from having raw and separate ERS- and AIS datasets, to achieving a dataset containing assembled AIS tracks for each fishing trip, are illustrated in Figure 5.3. The following sections describe each of the steps of this process in more depth.

5.2.1 Matching ERS- and AIS Messages

The start time of an AIS track for a vessel is defined as the time when the vessel is done with its last fishing haul. The stop time for the AIS track is defined by the formula in Equation 5.1. The start- and stop time define the time-interval where the vessel is steaming towards a port, and AIS messages transmitted by the vessel in that time period are collected and sorted by time to construct the AIS track of the vessel. Because ERS uses the radio call sign as the identifier of who sent a message, and AIS uses MMSI, the radio call signs of the vessels are translated into MMSI during this matching process. A lookup table is built and used during this process to optimize the performance of repeated lookups.

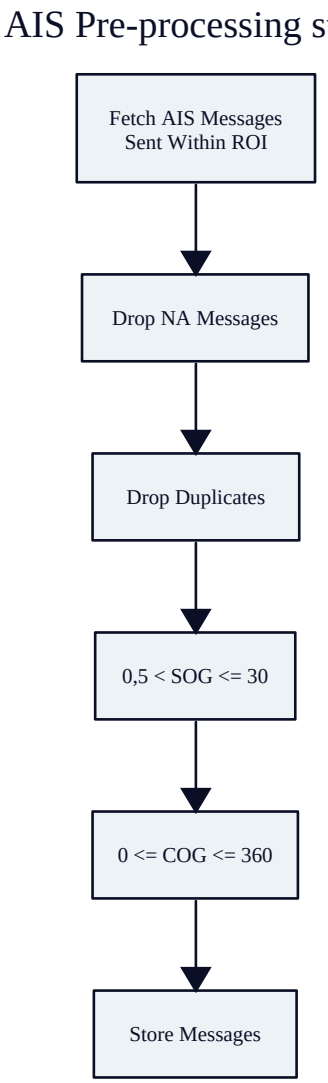


Figure 5.2: Diagram of AIS preprocessing steps

$$\text{mid-timestamp} = \frac{\text{<POR estimated timestamp>} + \text{<DEP timestamp>}}{2} \tag{5.1}$$

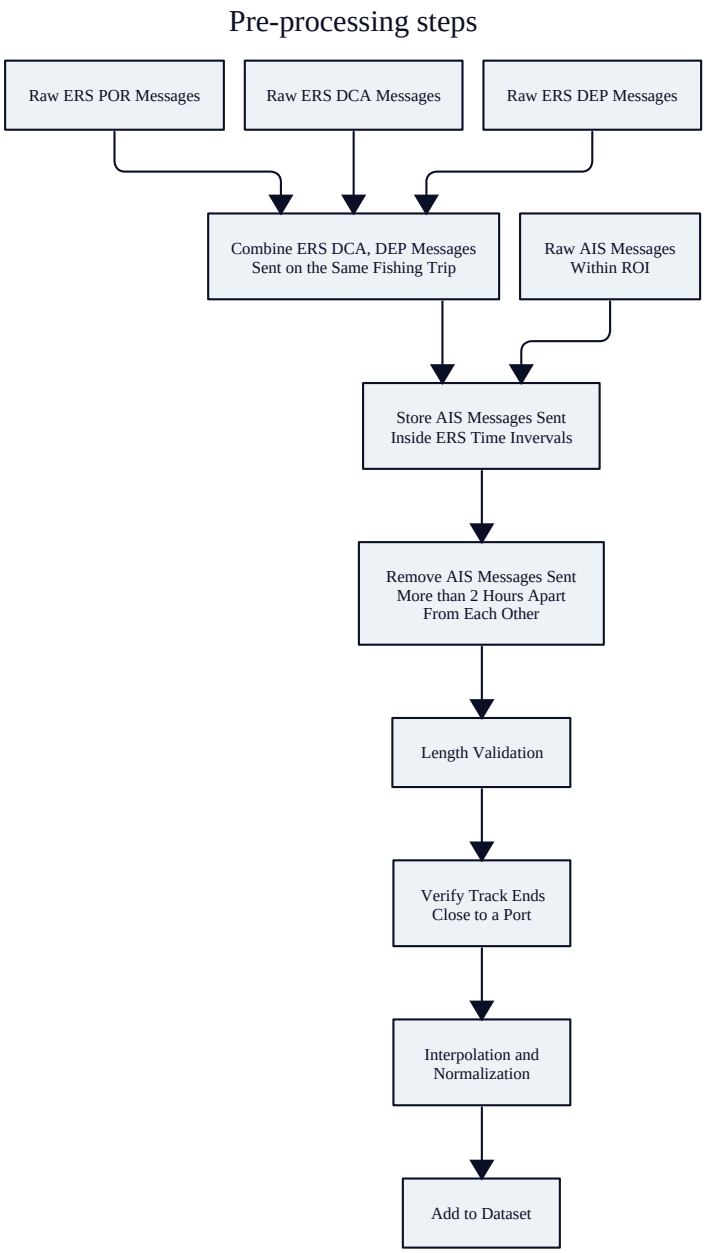


Figure 5.3: The preprocessing steps to create the final dataset

ERS fishing trips which does not match with any AIS messages are removed from the dataset, as the vessel has been fishing and delivering its catch outside the ROI when this occurs. The remaining ERS and AIS messages in the dataset will thus be messages which have been transmitted during trips inside our ROI. However, not all of these trips end at a port located inside our ROI, but rather pass through our region on its way to another port. Therefore, if the last AIS message of the fishing trip does not end within a five-kilometer radius of a port within our ROI, it is reasonable to expect that the vessel were passing through our region to deliver fish somewhere else.

5.2.2 Removing Outlier Messages

Due to reasons such as bad AIS signal, the estimated time of arrival being off, or the speed of the vessel being below 0.5 knots in the middle of the track, there can be a time span larger than two hours between two succeeding AIS messages. This makes the interpolation of the track, which happens at a later stage of preprocessing, remove the track, because the gap is deemed too large to perform interpolation. Often, this gap occurs early in the track, or after the vessel has reached the port, meaning that the tracks are still valid. To prevent the unnecessary reduction of the size of our dataset, which reduces our dataset almost in half, a preprocessing step was added to reduce the number of tracks removed because of two hours gaps.

If the time-interval between two AIS messages are more than two hours, and the messages are located in the middle of the AIS track, it will most likely not pass the length validation in the next preprocessing step if being cut. However, if this occurs with messages on either end of the track (at the start or the end), the remaining messages may still form a track which is long enough to pass the length validation. Therefore, if at any point in the track the time-interval between two AIS messages is more than two hours, the track is cut at that location.

When the two-hour gap is in the beginning of the AIS track, every message *before* the gap is removed. When this gap is located at the end of the AIS track, it is not suitable to remove every message before the gap, as the vessel might already be close to the port. In that case, the part of the track *after* the gap is removed instead, leaving the part of the trajectory where the vessel is heading from its last fishing location towards shore.

There is still some loss in number of tracks compared to the starting amount due to the gaps, because some tracks will be too short when the gap is around the middle of the track (where cutting the track makes it shorter than 4 hours, or 20 messages, which is the minimum requirement to pass the length-validation).

The number of tracks removed by interpolation is however significantly smaller after introducing this step, compared to before it being added.

5.2.3 Track Verification

AIS tracks should not be too short when used during training, as longer tracks provide more value in regard to the navigational behavior of fishing vessels. The requirement used in [20] is that a trip should be at least 4 hours long, and consist of minimum 20 AIS messages. This requirement is also applied to our dataset, meaning that AIS tracks which does not fulfill this requirement are removed. While this reduces the size of our dataset, it still retains a substantial enough size to train a transformer model with.

The last AIS track verification before a track is considered valid, and added to the final dataset, checks if the AIS track ends within five kilometers of any port in our ROI.

5.2.4 Interpolation and Normalization

Interpolation is performed to even out the frequency of AIS messages in a single track. The interpolation process is the same as the one used during preprocessing in GeoTrackNet[22, 23] and TrAISformer[20]. The code for the interpolation process is available on GitHub[2][3]. The tracks are interpolated such that there is a five-minute interval between each AIS message in the track. Because of the high frequency transmission rate of AIS, especially close to shore, the interpolation reduces the total number of messages in the track.

For the TrAISformer model to be able to use the AIS tracks, the latitude, longitude, speed over ground, and course over ground in the AIS messages must be normalized. The min-max normalization method defined in Equation 5.2 is used to normalize the values, because all the minimum- and maximum values are known prior to normalization. The min- and max value of the latitude and longitude is constricted to the coordinates of the polygon (bounding box) which the AIS messages are retrieved from, the speed is between 0 and 30, and the course of a vessel should only be between 0 and 360 degrees.

2. https://github.com/CIA-Oceanix/GeoTrackNet/blob/master/data/dataset_preprocessing.py#L204-L222
3. https://github.com/CIA-Oceanix/GeoTrackNet/blob/master/utils.py#L192-L239

$$\text{min\_max}(x) = \frac{x - \text{min\_val}}{\text{max\_val} - \text{min\_val}} \tag{5.2}$$

5.2.5 Dataset Format

Listing 5.1 is an example of the format of the dataset after preprocessing. Each element in the list contains information about a single trip by a vessel. The *mmsi* and *traj* entries in the dictionary are required by the TrAISformer model during training, with the latter consisting of a list of normalized AIS messages (latitude, longitude, SOG, and COG). The *port* entry in the dictionary maps to a tuple with the coordinates of the port the vessel headed to for that particular track, and the haversine distance[30] between the last AIS message and the closest port in kilometers. The *port* entry is used solely for evaluating the port-prediction algorithm, and not when predicting the trajectories of the vessels.

```
[
    {
        "mmsi": 123456789,
        "traj": [(AIS Message 1), ..., (AIS Message N)],
        "port": ((lat, lon), <distance to port>),
    },
    {
        ...
    }
]
```

Listing 5.1: The format of the dataset

5.3 Destination Port Prediction

The destination port of a fishing vessel is forecast based on the predicted trajectories output from the TrAISformer deep learning model. This section delves into the implementation details of how the destination port is chosen, based on these predicted trajectories. Figure 4.7 from Chapter 4 shows an overview of all the steps taken to determine a destination port.

5.3.1 Converting the TrAISformer Output Values

The N trajectories which the TrAISformer model predict for a single vessel is output as a time series with the AIS attributes: longitude, latitude, speed over

ground (knots), and course over ground.

The data points are ordered sequentially in time, such that the first data point is the predicted AIS message five minutes after the last AIS message in the input, the second data point ten minutes after, etc. The attributes for each data point is normalized, thus Equation 5.3 is used to denormalize the min-max normalization applied to all four AIS attributes.

$$\text{denorm(x)} = x(\text{max\_val} - \text{min\_val}) + \text{min\_val} \tag{5.3}$$

5.3.2 Smoothening the Predicted Trajectory with Moving Average

Because the data points of the predicted trajectory for a vessel are not as smooth as real trajectories, smoothening is applied to the coordinates. This aims to reduce the distance from potential outliers in the predicted trajectory, allowing the port prediction algorithm to focus on the overall directional heading of the predicted trajectory rather than potentially being affected by a single outlier which accidentally fall within the radius of another port's area.

A trajectory consists of a sequence of 2D coordinates, SOG and COG. Only the coordinates are targeted by the smoothening. Before averaging the coordinates, the trajectory is extended by duplicating the edge coordinates by the floor value of $\frac{\text{window\_size}}{2}$. The window size is the total number of coordinates included in the calculation of each coordinate's new value. We use a window size of 5, as it leads to trajectories being more natural, while not completely removing the smaller turns.

Once the trajectory has been extended by duplicating the edge values, the x- and y-values are split into individual arrays. The discrete, linear convolution is then calculated on both arrays using NumPy's *convolve*[4] function. The convolution operation requires two one-dimensional arrays: the first is the array of values to be smoothened, and the second is an array of weights, corresponding to the size of the sliding window. Since all coordinates in a trajectory should be equally weighted, the weights are set to $\frac{1}{\text{window\_size}}$.

During the convolution, each coordinate in the x- and y-arrays is replaced

4. `https://numpy.org/doc/stable/reference/generated/numpy.convolve.html`

by the average of its neighboring coordinates within the window. The result is a sequence of smoothened coordinates for the x- and y-values, which are recombined to restore the coordinates of the trajectory. This version of the trajectory will be more smooth than it initially was. The size of the sliding window can be adjusted to control the level of smoothing: increasing the window size results in a smoother trajectory, while decreasing it results in a rougher trajectory.

5.3.3 Predicting the Destination Port for Each Track

For a single predicted trajectory from the TrAISformer model, each data point the trajectory consist of is iterated through when the port the trajectory is heading for is being decided. This means that the position of the vessel is checked every five minutes. The haversine distance formula[30], defined in Equation 5.4, is used to calculate the distance d between the predicted position of the vessel and each of the available ports. The distance is used to determine if the vessel is within the defined circular area of any port. If a vessel in within the area of multiple ports, the distance is used to decide which one the vessel is the closest to.

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\text{lat}_2 - \text{lat}_1}{2}\right) + \cos(\text{lat}_1)\cos(\text{lat}_2)\sin^2\left(\frac{\text{lon}_2 - \text{lon}_1}{2}\right)}\right) \tag{5.4}$$

The radius of the area surrounding each individual port is set to two kilometers initially. This means that for a port to be considered as a destination port by the port prediction algorithm, the predicted trajectory must be within two kilometers of it (in addition to having a predicted speed less than 0.6 knots).

The radius is incremented by one kilometer if no port is predicted for the current radius (i.e., if no port is predicted when the radius is two kilometers, meaning that no part of the trajectory where the speed is less than 0.6 knots is within this range of any port, the algorithm repeats using a radius of three kilometers, etc.). The maximum radius is set to five kilometers. If no port is predicted when this maximum radius is reached, the algorithm doesn't make a prediction, as the vessel is most likely more than 8.5 hours from shore, or taking an abnormal path which it cannot make an accurate prediction for.

5.3.4 Assigning Probabilities for each of the Ports

The TrAISformer model is set to predict $N = 16$ trajectories for a single input. Each trajectory has a destination port predicted, unless it is decided that the uncertainty is too large to predict a port, which leads to a none-prediction.

The probability of a port is decided based on how many of the trajectories were predicted to visit that port, given by Equation 5.5.

$$\text{prob(port)} = \frac{\text{n\_visited}}{\text{N}} \tag{5.5}$$

If 9 of the 16 trajectories had port X predicted as the destination port, the probability of that port being the vessel's destination port is $\frac{9}{16} = 0.56$. The sum of the probabilities for each port is always equal to 1. It is possible to configure the TrAISformer model to predict more trajectories than 16, which would give a more fine-grained overview of the port probabilities, due to each predicted trajectory having less of an impact on the overall probability of the port it heads for.

5.4 Web Application for Visualization of Data

To gain a deeper understanding of the data being worked with, a simple web application was developed using Flask[12] and Folium[10]. This application's primary function is to visualize predicted vessel trajectories, true vessel trajectories, and ports in an interactive map.

When the server receives a request for a trajectory, the index of the trajectory in the test set and the length of the input to the TrAISformer model is given. Subsequently, the port prediction algorithm processes the output trajectories from the TrAISformer model, enabling the visualization of port probabilities, and the predicted destination port, in the map.

Before being plotted into the map, the trajectories are denormalized and smoothened. In the map, the input provided to the TrAISformer model, along with the predicted trajectories and the ports with a non-zero probability, are visualized. Because the TrAISformer model always predicts 8.5 hours in advance of the last input received, any trajectory segments beyond port arrival are omitted to reduce noise in the map.

Every figure which includes a map in this thesis is created using this web application, with the figures from Section 6.5 in Chapter 6 showing the full capability of it.

6 Evaluation

This chapter presents the performance of the two models used in this thesis, as well as the port prediction algorithm. A comparison is also made between the baseline model and the approach consisting of the TrAISformer model and port prediction algorithm. Visualization examples of the port predictions using the latter approach are also presented, with the varying cases that may arise.

6.1 Setup

6.1.1 Dataset Details

The dataset used for training, validation, and testing comprises 361, 284, and 280 unique vessels, respectively. Vessels from one subset may occur in the other datasets as well, but all the AIS tracks are unique (e.g., a vessel may have a track in the training set, while also have another track in the validation or test set). The original dataset was split into training, validation, and test sets in sizes of 80%, 10%, 10%, containing 13735, 1716, and 1718 AIS tracks. The dataset contain a total of 370 unique fishing vessels.

The number of ports each vessel has visited varies, with the distributions shown in Figure 6.1, Figure 6.2, and Figure 6.3. The x-axis represents the number of unique ports a vessel has visited, while the y-axis represents the number of vessels that have visited the same number of unique ports. For example, from

the figure of the histogram from the training data, Figure 6.1, 47 vessels have visited only a single port (one or more times).

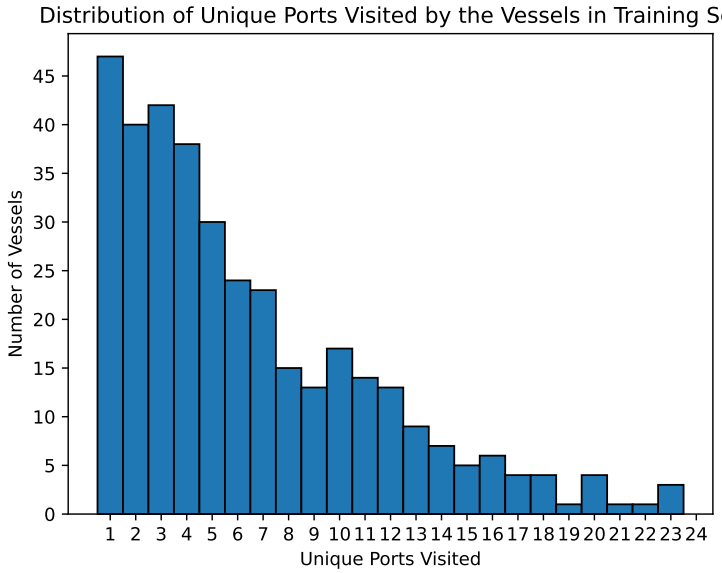


Figure 6.1: Histogram of number of unique ports visited by vessels in training set

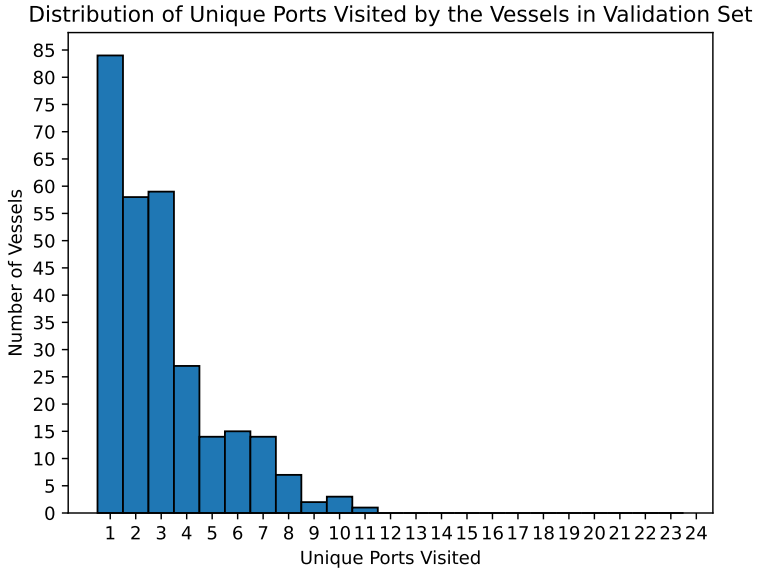


Figure 6.2: Histogram of number of unique ports visited by vessels in validation set

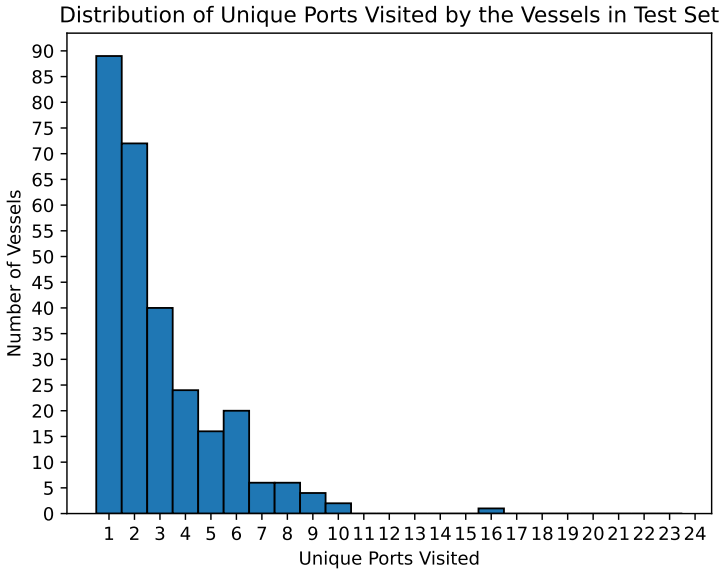


Figure 6.3: Histogram of number of unique ports visited by vessels in test set

6.1.2 Hardware

The measurements from this chapter were conducted on an Ubuntu 22.04.3 LTS system equipped with an Intel(R) Xeon(R) CPU E3-1275 v6 @ 3.80GHz CPU, and an NVIDIA GeForce GTX 1080 Ti GPU with 12 GB of memory. The GPU operates with CUDA version 11.4, which necessitates the use of PyTorch version 2.1.2 for compatibility.

6.2 Baseline Model

The following section highlights the performance of the baseline model in terms of prediction accuracy and time usage. The port prediction accuracy is measured using two different metrics, while the time usage is measured during training, evaluation, and making a prediction.

6.2.1 Port Prediction Accuracy

Table 6.1 shows the accuracy of the baseline model on the test set. The *Most Probable Port* column represents the accuracy of the model where it correctly predicts the destination port of the vessel, meaning that the most probable

	Most Probable Port	Top Three Most Probable Ports
Accuracy	0.4796	0.7328
Correct	824	1259
Total Predictions	1718	1718

Table 6.1: Baseline model port prediction accuracies

port, according to the model, was the actual destination port of the vessel for that trip. The second column, *Top Three Most Probable Ports*, shows that in 73% of the test cases the correct destination port of a vessel is amongst its top three most probable ports, according to the model. This means that in approximately 26% of the cases, the vessel's destination port was its second or third most visited port.

Out of the 1718 tracks in the test set, the baseline model predicted the wrong port in 893 cases, affecting 231 out of 280 different vessels. This means that only 49 vessels always chose the port they have visited the most in the past as their destination port during the test. This does however not necessarily mean that all of them only deliver their catch at a single port, but that delivering catch at the other ports is less frequent, and those tracks not necessarily occurring in the test set.

While Table 6.1 presents the prediction results on the primary dataset, which is used by the deep learning model + port prediction algorithm, the baseline model was tested on two additional datasets. The two datasets are re-shuffled versions of the AIS tracks located in the primary dataset, such that tracks that are in the training set of the primary dataset might be in the validation set or test set instead for these datasets. The proportions of the training, validation and test set remain the same as the primary dataset, and the baseline model is retrained when measuring its performance on the two alternative datasets. The results are shown in Table 6.2, and shows that the accuracy of the model only varies slightly.

6.2.2 Time Usage

Because the baseline model is simplistic, both training and making predictions with is fast. The time usage, shown in Table 6.3, was measured by running each operation (training, evaluating, and predicting) five times, and calculating the average time spent.

The measured times for all three operations of the model demonstrate that

	Most Probable Port	Top Three Most Probable Ports
Accuracy	0.4913	0.7352
Correct	844	1263
Total Predictions	1718	1718
	Most Probable Port	Top Three Most Probable Ports
Accuracy	0.4889	0.7276
Correct	840	1250
Total Predictions	1718	1718

Table 6.2: Baseline model port prediction accuracies on the test-dataset, and a re-shuffled version of the same dataset

	Training	Evaluation	Predict a Single Port
Number of Tracks	13735	1718	1
Mean Time (ms)	8.6543	2.1578	0.0051

Table 6.3: Time usage of the baseline model to train, evaluate, and predict

the simplicity of the model offers greater speed than more complex solutions, due to the lack of heavy computation required. Also, no special hardware is required to run this model in a real-time environment. The fast training time allows for the model to be continuously updated, as retraining is a fast operation to perform. Additionally, measurements indicate that predicting the destination port of a vessel is done instantly, while being correct approximately 50% of the times.

6.3 TrAISformer Trajectory Prediction

In this section, the TrAISformer deep learning model is evaluated. The accuracy of the predicted trajectories is measured using varying input lengths, as well as the time usage to compare against the baseline model. These measurements do not include port predictions, only trajectory predictions.

6.3.1 Mean Trajectory Prediction Accuracy

Figure 6.4 presents the accuracy of the TrAISformer model on our dataset, with the accuracy given as the mean error of all 16 predicted trajectories for each input track. The lines are color-coded based on the time span of the input. An input length of 0.5 hours provides the model 6 AIS messages of input

before making the prediction, while an input length of 4 hours provides 48 AIS messages in this time span (5-minute time interval between each message, due to interpolation).

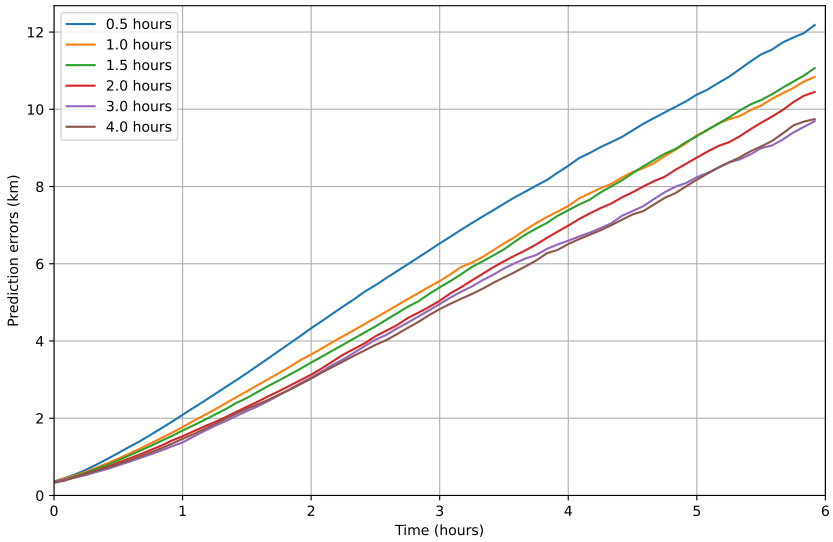


Figure 6.4: Mean prediction accuracy for trajectory predictions with various input lengths

The graph shows that the model predicts more accurately with more input data, as expected. The largest improvement occurs between 0.5 hours of input and 1 hour of input, indicating that less than 1 hour of input is insufficient for making trajectory predictions more than 1.5-2 hours ahead in time.

Comparing our mean accuracy with the mean accuracy of the original dataset used in TrAISformer paper[20] reveals that the model perform better on our dataset. For their dataset, using one hour of AIS input, the mean errors of the model were 4.44 km, 9.07 km, and 14.80 km for 1, 2, and 3-hour ahead predictions (during our re-run, without using the best-of-N principle when calculating the mean error), respectively. The only change between the two evaluations is the dataset, as the transformer model is unchanged. Our training set contains approximately 3500 more AIS tracks, and our ROI is larger, being 307 490 km^2 compared to approximately 47 000 km^2 in the original dataset. Exploring the impact of the ROI is left as future work.

6.3.2 Time Usage

The time usage has been measured for training and predicting different number of trajectories with the TrAISformer model. Table 6.4 presents the measured times. Training takes a little over two hours on the hardware specified in Subsection 6.1.2 with our dataset. For all tests, three measurements were taken, with the presented results are the average times. The prediction times do not include the time taken to load data between the CPU and GPU.

	Training	Predict Trajectories	Predict Trajectory
Number of Tracks	1718	16	1
Mean Time (s)	7436	9.231	0.800

Table 6.4: Time usage of the TrAISformer model to train, evaluate, and predict

While the TrAISformer model uses 9 seconds to predict 16 trajectories for a single input, it is still feasible to use in a production environment for the controllers. The decision of which port should be controlled, because a vessel is heading there, do not need to be made within seconds. This allows for the use of methods which do not provide results in sub-second times. Retraining the model is also not very time-consuming, and can also be improved by using newer hardware.

6.4 Port Prediction Algorithm

The performance of the port prediction algorithm is measured and reported in the following sections. The accuracy of the model is measured, and compared against the baseline model, as well as the time usage.

6.4.1 Accuracy

The prediction accuracy port prediction algorithm, using trajectories predicted by the TrAISformer model, is shown in Figure 6.5. The input lengths used for the TrAISformer model are 0.5 hours, 1 hour, 1.5 hours, 2 hours, 3 hours, and 4 hours, which corresponds to 6, 12, 18, 24, 36, and 48 AIS messages of input. For each input track from the test dataset, 16 trajectory predictions are generated. These trajectories are passed to the port prediction algorithm, which predicts the destination port based on the behavior of the predicted trajectories.

The accuracies in Figure 6.5 indicates how often the destination port predicted as the most probable one corresponds to the actual destination port of the

vessel. Figure 6.6 shows how frequently the actual destination port is among the top three most probable ports (instead of just the most probable one) for the baseline model and the TrAISformer model + port prediction algorithm.

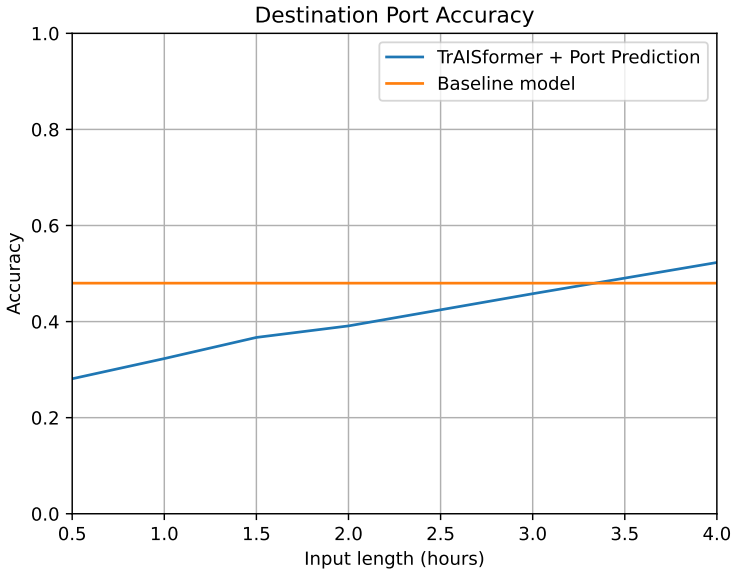


Figure 6.5: Port prediction accuracy comparison with baseline model

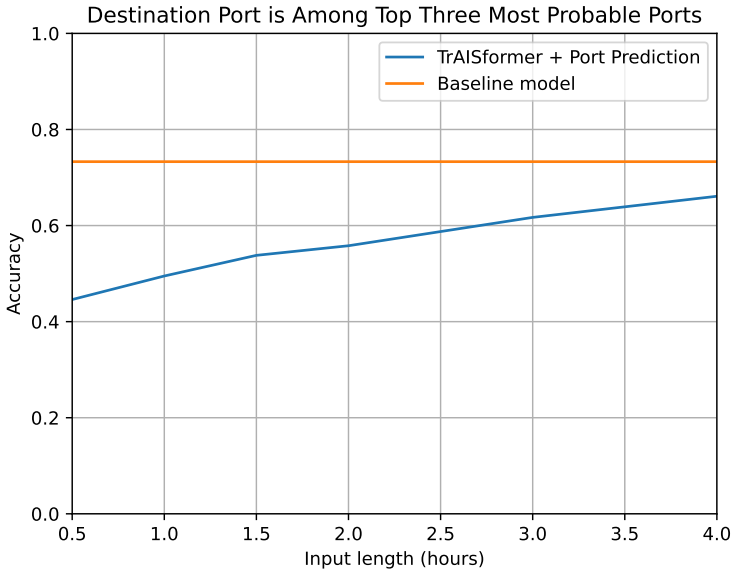


Figure 6.6: Accuracy for the true destination port being among the top three most probable ports

Both graphs demonstrate that the accuracy of the combination of the transformer model and the port prediction algorithm increases with the length of the input. The accuracy of the baseline model remains constant because it solely relies on a vessel's MMSI as input, and not AIS. The baseline model performs better when there are fewer AIS messages available to use as input for the TrAISformer model, as its predicted trajectories become more inaccurate. These low-input scenarios favor the baseline model, as it relies on the past history of the vessel, while scenarios with high amounts of input available favors the TrAISformer model.

In both graphs, incorrect destination port predictions for a given input stem from two main reasons: either the model predicts the wrong destination port, or it is not confident enough to predict a port. In about 8-13% of port predictions do not result in any predicted port (depending on the input length, where more input results in less occurrences of no ports being predicted), slightly impacting the accuracies showed in both Figure 6.5 and Figure 6.6. The results excluding cases where no port is predicted, focusing solely on correct or incorrect predictions, are depicted in Figure 6.7 and Figure 6.8.

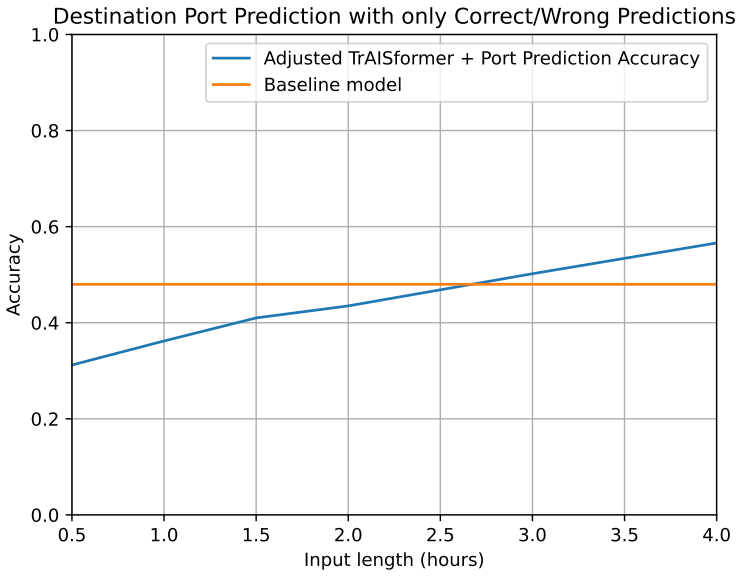


Figure 6.7: Port prediction accuracy with only correct/wrong predictions

6.4.2 Time Usage

The time usage of the port prediction algorithm is presented in Table 6.5. Measurements were conducted three times, and the average times are reported.

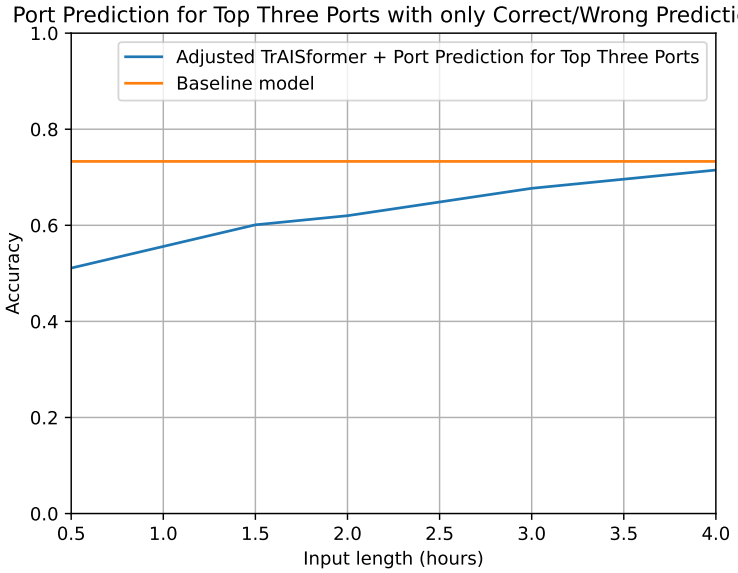


Figure 6.8: Accuracy for the true destination port being among the top three most probable ports, with only correct/wrong predictions

In the first measurement, the algorithm predicts a destination port using the 16 trajectories generated by the TrAISformer model for a single vessel. The reported mean time is 37 ms for the port prediction, although this value is highly dependent on how much the radius of the area surrounding the ports have to be increased. If a destination port is chosen early in the algorithm, because the trajectories approach close to the location of the port (hence not having to increase the radius), the prediction time is typically around 15-20 ms. Conversely, if no port is predicted, the average prediction time increases to about 50 ms.

	Predict Destination Port	Predict Destination Port
Mean Time (ms)	37.078	9.551
Number of Trajectories	16	1

Table 6.5: Time usage of the port prediction algorithm

The prediction times are fast, because the port prediction algorithm does not involve any complex computations. In comparison to the time taken by the TrAISformer model, from which this algorithm receives its trajectories to predict for, the prediction of a destination port is almost instantaneous.

6.5 Visualizations of Port Predictions

The following figures illustrate the AIS input to the TrAISformer model, the predicted trajectories, and the probable destination ports for the predicted trajectories. Each figure follows the following scheme:

- The input to the TrAISformer model is colored with a white line.
- The red trajectory is the true trajectory of the vessel.
- Trajectories not colored red or white are the trajectories predicted by the TrAISformer model.
- Blue marker indicates the predicted destination port, and is based on the predicted trajectories. The probability of that port being the destination port is attached to the marker.
- Red marker indicates the actual destination port of the vessel, also attached with the predicted probability of it being the destination port. If there is no red marker, the correct destination port was predicted, and the marker is instead colored blue. Red markers will have a surrounding circle indicating a 5 km radius of the port.
- Gray marker indicates ports which are not the most probable, not the actual destination port, but has a non-zero probability of being the destination port.

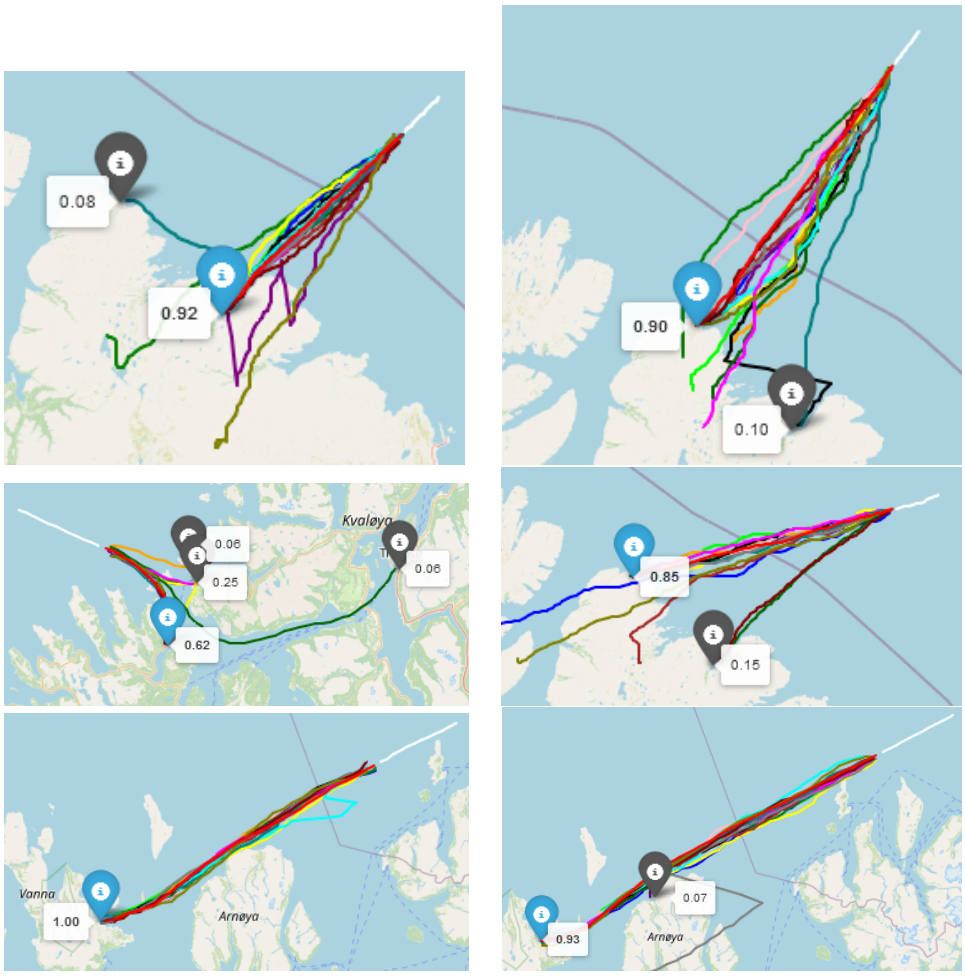


Figure 6.9: Correct predictions when using an input length of one hour

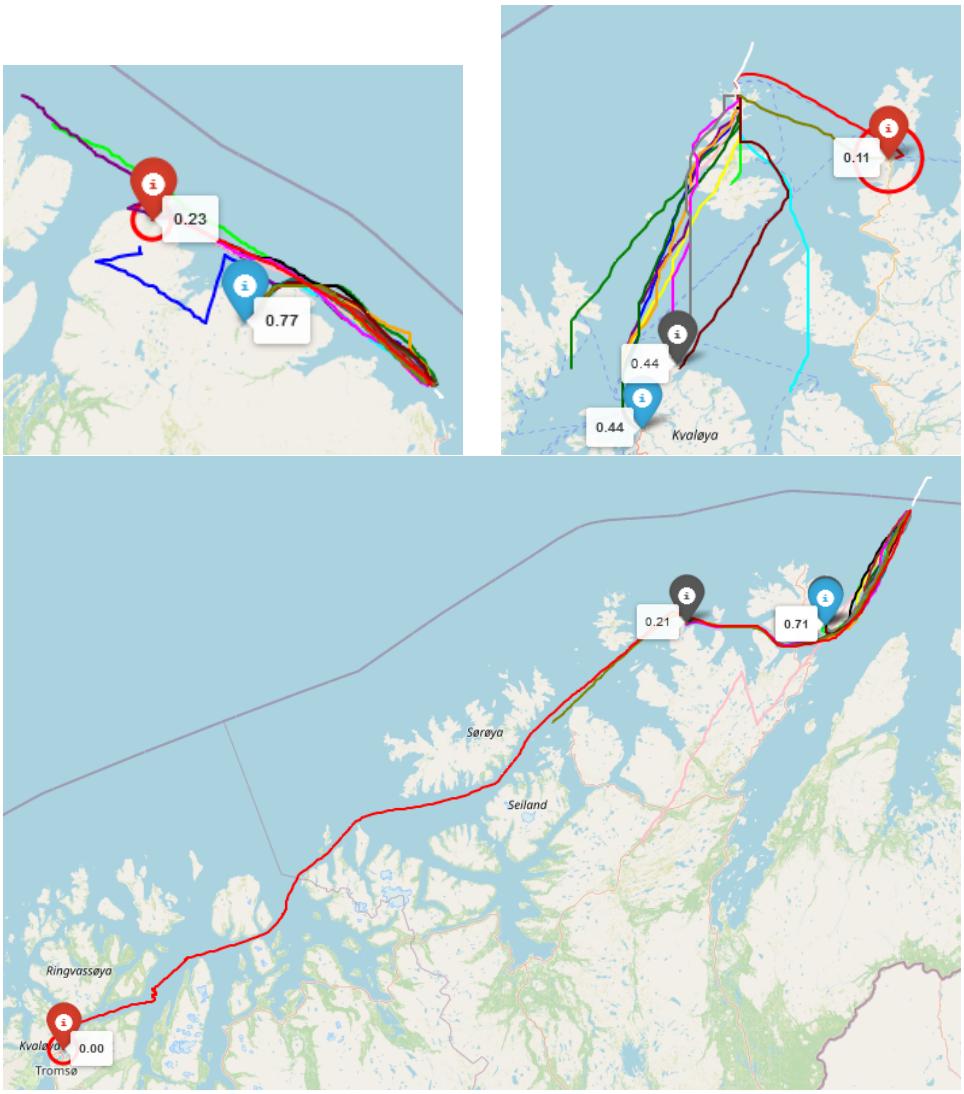


Figure 6.10: Wrong predictions when using an input length of one hour

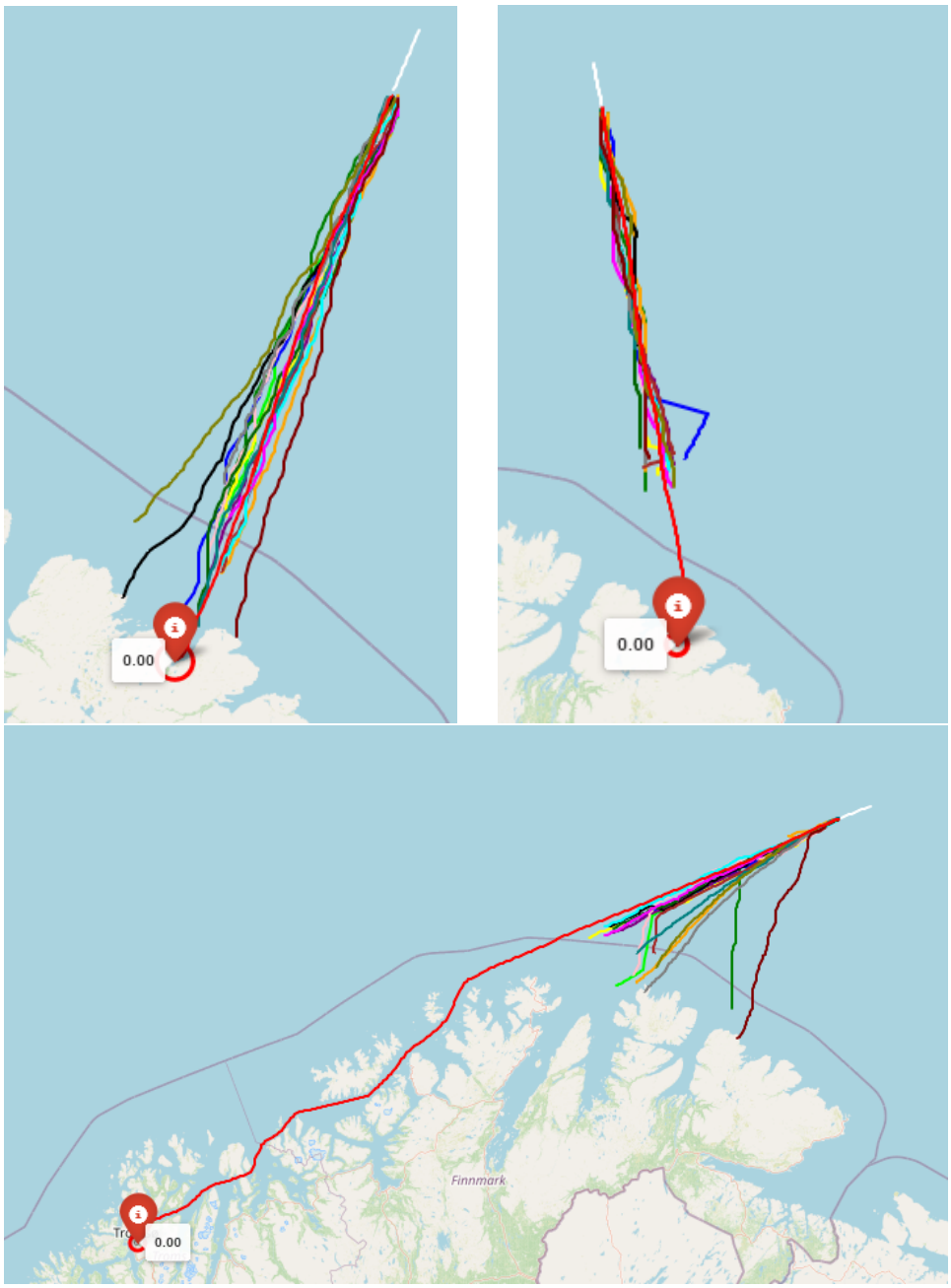
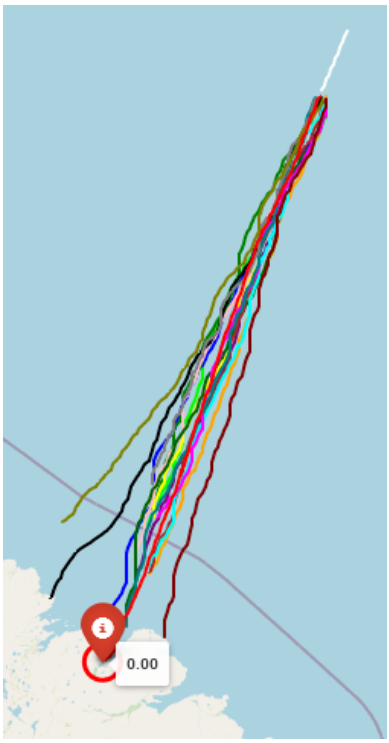
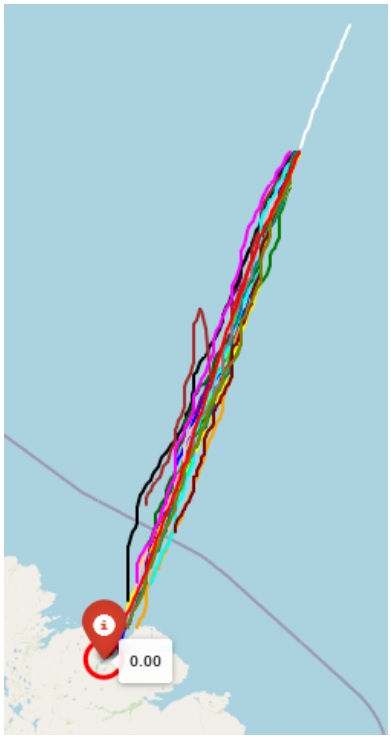


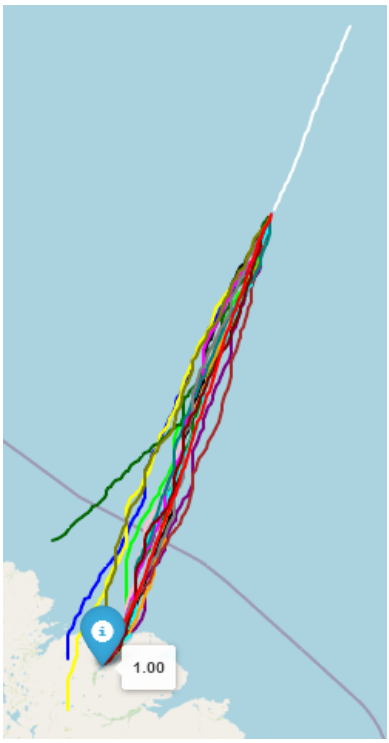
Figure 6.11: No predictions when using an input length of one hour



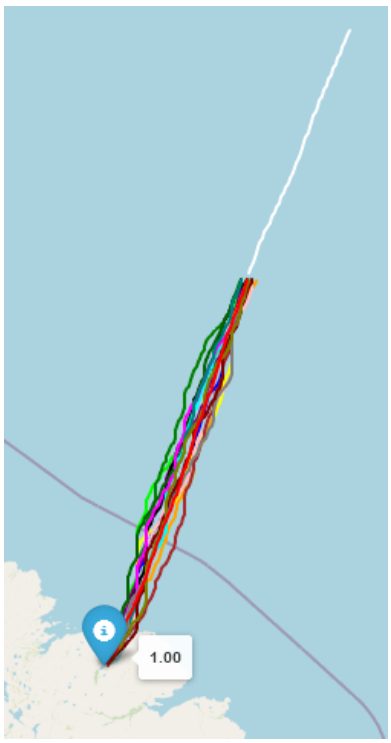
(a) One hour of input



(b) Two hours of input



(c) Three hours of input



(d) Four hours of input

Figure 6.12: Example where increasing the amount of input eventually leads to a correct port prediction

7 Discussion

This chapter discusses alternate methods to improve the proposed solutions in regard to predicting the destination port of fishing vessels. It also covers issues and improvements to the current models, as well as possible directions forward to improve their performance.

7.1 AIS vs. VMS

An alternative to AIS is Vessel Monitoring System (VMS), which is designed with the purpose of monitoring vessel activity. In Norway, vessels transmit VMS messages every 10 minutes, with these messages containing information about the vessel's GPS position, speed over ground, and course over ground. The messages transmitted by Norwegian fishing vessels longer than 15 meters are available through the Directorate of Fisheries[31].

In addition to the difference in transmission rate between AIS and VMS, the GPS coverage between the two systems may vary[32, 33], especially when a vessel is far out at sea. During our preprocessing stages, gaps larger than two hours were observed in the data, requiring the AIS tracks to be shortened before interpolation were performed.

Because of the similarity between AIS and VMS, and since Norwegian fishing vessels longer than 15 meters are required to use both systems[3, 34], it is possi-

ble to combine the two systems to cover the gaps that can occur from using only AIS[33]. Another possibility would be to use VMS instead of AIS to improve coverage. However, this would reduce the potential for improving vessel trajectory predictions by incorporating additional information into the deep learning model, as VMS contains less information than AIS. Subsection 7.4.3 discusses some additional features which can be incorporated into the models.

7.2 Baseline Model

The baseline model has some advantages and disadvantages when it comes to predicting the destination ports of fishing vessels. Due to its simple implementation, the model is fast to both train and make port predictions. It is also easy to use, as the only requirement for the model is the MMSI of the vessel for which it should make a prediction. Moreover, it is more accurate than our initial estimates.

One reason for its better-than-expected accuracy is that fishing vessels, particularly smaller ones around 15 meters in length, may have a limited selection of ports they consider for delivering their catch. While there are 68 possible ports in our dataset, these vessels might favor ports which are closer to their home area. Additionally, these smaller vessels will also have a lower fish storage capacity compared to larger fishing vessels, resulting in more frequent and shorter fishing trips. This can cause a skew in the dataset which favors the smaller vessels, thereby improving accuracy of the baseline model due to the limited port selection.

Vessels capable of visiting many ports are one of the major weaknesses of the baseline model. The model always assigns the highest probability to the port the vessel has visited the most in the past. For vessels that distribute their visits somewhat evenly across multiple ports, the model will struggle to achieve a high accuracy in predicting the destination port. The model is also not able to predict destination ports of vessels not occurring in its training set.

7.3 Port Prediction Algorithm

The port prediction algorithm utilizes most of the information provided by the TrAISformer model, excluding the predicted course over ground. Since the algorithm relies solely on the predicted trajectories and speed to forecast the destination port, its accuracy is tied to the TrAISformer model output. When the TrAISformer model accurately predicts the trajectories, the port prediction

algorithm is more likely to predict the correct destination port.

However, if the predicted trajectories from the TrAISformer model deviate from the vessel's actual course, the port prediction algorithm cannot adjust accordingly. Consequently, the algorithm gives probabilities for different ports rather than solely identifying the most likely port to be visited. This approach acknowledges that while the predicted trajectories may not always be accurate, they represent the most probable parts based on the given input. Increasing the input to the TrAISformer model enhances the accuracy of the port probabilities by improving the quality of the predicted trajectories. This can be seen in the results from Chapter 6, where the accuracy increased when the input increased.

One potential issue can arise based on how the destination port of a vessel is predicted. The TrAISformer model consistently forecasts trajectories 8.5 hours ahead from the last received AIS message in the input. In some cases, this may result in parts of the trajectories intersecting with land. When the port prediction algorithm iterates over the predicted vessel position in a trajectory, it does not differentiate whether the vessel is on land or in water. Consequently, the trajectory may have crossed land before arriving at a port, as illustrated in Figure 7.1.

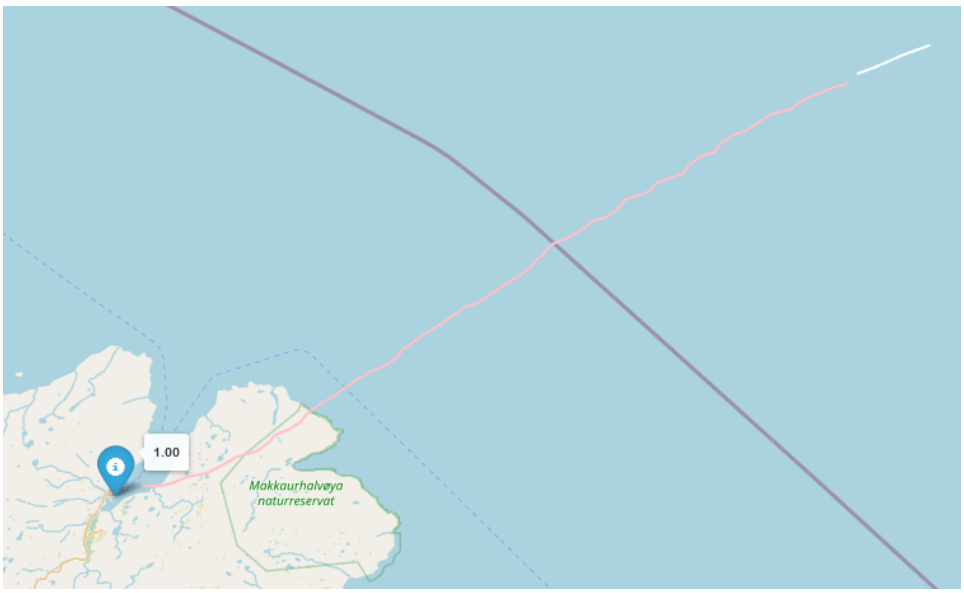


Figure 7.1: Example of a destination port being predicted after the vessel has crossed land

It could be beneficial to incorporate a step into the port prediction algorithm which prevents trajectories that crosses land to affect the overall probabilities

of ports being the destination port. The added step could cut off the trajectory at the point it intersects with land, causing only the prior part of the trajectory to be utilized when predicting a port.

Initially, machine learning approaches were considered for predicting the destination port of vessels, based on previous work with AIS data[19, 35], instead of the chosen algorithm. However, due to factors such as time, and observing that the predicted trajectories from the TrAISformer model often closely approaches the ports, the chosen algorithm was deemed sufficient for predicting destination ports.

While using a more advanced port prediction method has the potential to improve accuracy, it is not currently considered as the main improvement area of our approach, as it heavily relies on the predicted trajectories. Instead, the primary area of improvement is considered to be predicting more accurate vessel trajectories. A possible approach to improve this, aside from acquiring more training data, is proposed in Subsection 7.4.3.

7.4 Future Work

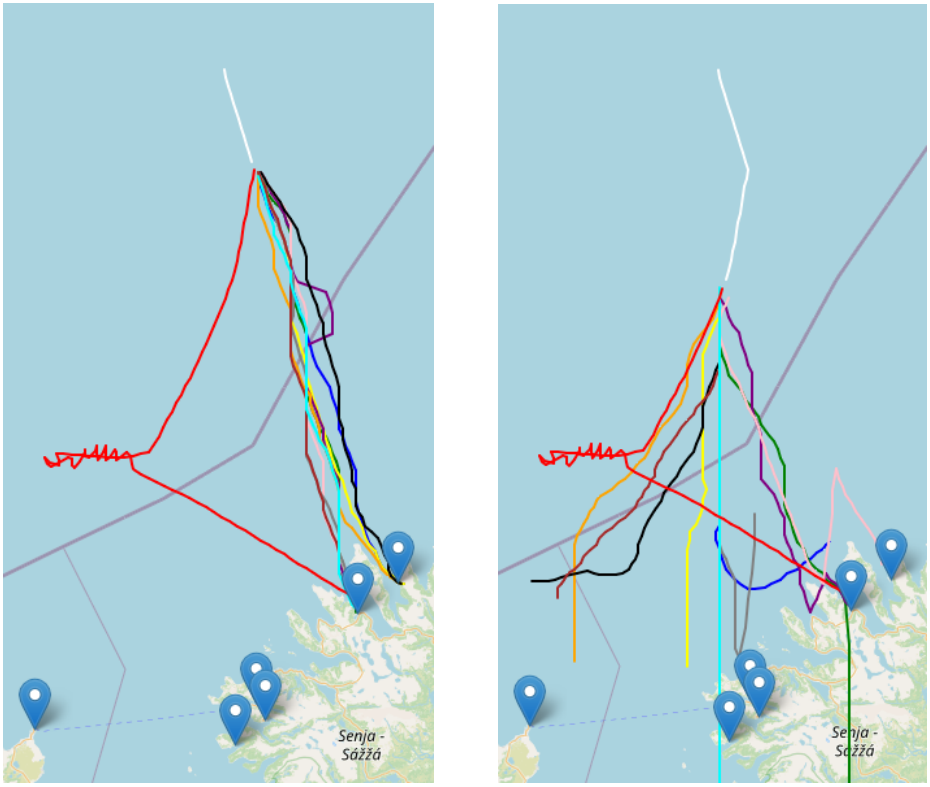
Although both the proposed solutions manage to predict the destination port of a vessel correctly in the majority of the cases, there are still improvements which are believed to improve their performance.

7.4.1 Using a Smaller ROI

As stated in Chapter 6, our selected ROI is significantly larger compared to the TrAISformer paper[20]. Their training set contains 770 000 AIS messages within a 47 270 km^2 ROI, whereas our training set contains 1 600 000 AIS messages within a 307 490 km^2 ROI. This results in AIS message densities of 16.27 messages per square kilometer for their dataset and 5.27 messages per square kilometer for our dataset.

While different densities in the number of AIS messages per square-kilometer do not necessarily affect the model's accuracy, it is not unlikely. The largest impact will likely be in regions where fewer vessels have traveled, as predicting the trajectory of a vessel is more difficult if no, or few, vessels have taken the same route. Figure 7.2 illustrates such a case. In the first image, the input to the model (white line) follows a relatively known route, while in the second image, the vessel takes an unexpected turn. In the first image, the input length is one hour, while in the second image, it is two hours. The unexpected turn

occurs because the vessel seems to be heading for more fishing (as indicated by its true trajectory, colored red), moving along a route that is not directed towards a port. Consequently, the model struggles to predict its trajectory accurately, resulting in predicted trajectories which does not make sense for a vessel heading towards a port.



(a) Vessel taking an expected route

(b) Vessel taking an unexpected route

Figure 7.2: Trajectory predictions for vessel before and after taking an unexpected turn

Smaller ROIs might yield more accurate predictions, if there is sufficient data for the area. However, in low-traffic areas, there might not be a significant improvement compared to using a larger ROI. The initial reason for choosing a larger ROI was to reduce the number of models required to cover the Norwegian coast. Investigating the optimal ROI size, and if the ROI can cover all of the Norwegian coastline, is left as future work.

7.4.2 Improve Trajectory Extraction using Fishing Activity Detection

While most of the AIS tracks in the dataset contain trajectories where the vessel is heading directly for a port, there are also a few tracks where this is not the case. Figure 7.2 is an example of such a track. Ideally, these tracks should not be included in the dataset, especially the training set, as the model may struggle to predict sensible trajectories. It is challenging for the model to learn the desired vessel behavior when they are not heading directly for a port, or engaged in activities such as fishing, as there is no consistent pattern to follow. Although it would be possible to manually review and remove these tracks, the process is time-consuming given that there are more than 17 000 tracks in the dataset.

To improve the quality of the dataset by removing trajectories not heading directly for a port during preprocessing, machine learning can be employed. Research papers have been published on detecting fishing activity based on AIS tracks[28, 29], utilizing the unique patterns of AIS tracks during fishing and steaming. This addition to the AIS preprocessing is left as future work.

7.4.3 Incorporating more Information into the Models

Both the baseline model and the TrAISformer model currently utilize the AIS data in different ways to predict the future behavior of a vessel. The available data is also utilized differently by the two models, although their goals are somewhat similar. The baseline model predicts a destination port immediately, while the TrAISformer model is one of two steps to achieve this task. However, there are still input features that can be added to both models, which are believed to improve their prediction accuracies. Exploring which input features yield the most improvement, or if they indeed enhance the models, is left as future work.

The baseline model currently relies solely on the past destination ports visited by a vessel to make predictions, disregarding the vessel's current location. Incorporating the vessel's location at the time of prediction is an essential step to explore for this model. It could prevent the model from predicting a destination port that is far from the vessel's current location, and enable it to instead predict the closer second- or third-most probable port. Additionally, other valuable information that could improve this model includes the vessel's fishing locations (e.g., which catch zone from Figure 7.3 that has been fished most recently in), the species targeted during fishing, and the departure port for the trip. Most of this information is available in the ERS messages transmitted by the vessel, but can also be extracted from AIS.

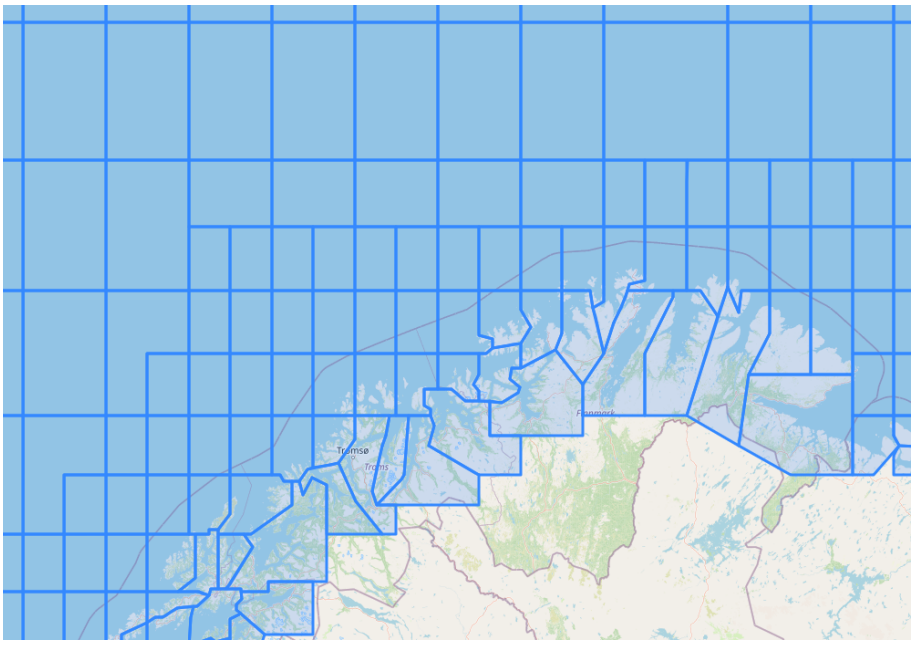


Figure 7.3: Catch zones from outside the coast of Norway

The TrAISformer model utilizes more of the available AIS data than the baseline model, by using AIS messages as input. This model is more generalizable than the baseline model, since it does not rely on the vessel's ID for predictions. Similar to the baseline model, it is possible to include information such as the vessel's recent fishing locations, the species targeted during fishing, and the departure port into the transformer model. Additionally, static vessel information, such as its length, found in the AIS messages, could also enhance the transformer model. This type of information may not impact the baseline model, as it already utilizes the vessel's ID.

8 Conclusion

This chapter outlines concluding remarks in regard to this thesis' problem statement, defined in Section 1.1 in Chapter 1, as well as our findings.

8.1 Concluding Remarks

The goal of this thesis was to investigate how to predict the destination ports of fishing vessels based on their AIS messages, as a first step towards a larger goal of developing a system which controllers can use to improve their ability to regulate the catch of these vessels. This thesis focused on researching and proposing approaches which can be further developed into such a system.

Two approaches utilizing the AIS of fishing vessels were proposed and compared against each other. Both approaches were trained on historical AIS trajectories within our defined region of interest in the northern part of Norway. The first approach, a statistical baseline model, based itself on the past port-visit history of each unique vessel to predict which port it is most likely to visit next, regardless of its position. The second approach, composing of a transformer deep learning model and a port prediction algorithm, used the transformer model to predict the most likely trajectories of the fishing vessel. These trajectories were used as input into the port prediction algorithm, which forecast the destination port of the vessel given these trajectories, as well as its probability.

Based on our results, the statistical baseline model performs better when predicting the destination ports of vessels which have a low variety in the ports they visit, but is tied down to only predicting destination ports of vessels it already knows. In comparison, the machine learning approach can predict the destination port of any vessel, regardless of it being in the training data or not, and has a better accuracy than the baseline model when there is more than three hours of AIS input available.

In conclusion, while both approaches predict the destination ports of vessels correctly in the majority of cases, the machine learning approach appears to be the most promising approach given its results and generalizability. This indicates that utilizing AIS and machine learning to forecast the destination port of a fishing vessel is a viable and effective method. Additionally, by incorporating more input features into both models, we expect their port prediction accuracies to improve greatly.

Bibliography

[1] Kystverket. *AIS Norway*. https://www.kystverket.no/en/navigation-and-monitoring/ais/ais-norge/. (Visited on 02/05/2024).

[2] NATO. *AIS (Automatic Identification System) overview*. https://shipping.nato.int/nsc/operations/news/2021/ais-automatic-identification-system-overview. (Visited on 02/05/2024).

[3] Norvald Kjerstad. *AIS Norway*. https://snl.no/AIS. (Visited on 02/05/2024).

[4] Norwegian Ministry of Trade, Industry and Fisheries. *Forskrift om posisjonsrapportering og elektronisk rapportering for norske fiske- og fangstfartøy (ERS-forskriften)*. https://lovdata.no/forskrift/2009-12-21-1743. 2009.

[5] Directorate of Fisheries. *Elektronisk rapportering fra fiskefartøy*. https://www.fiskeridir.no/Yrkesfiske/Rapportering-paa-havet/elektronisk-rapportering-ers. (Visited on 02/05/2024).

[6] Directorate of Fisheries. *Åpne data: elektronisk rapportering (ERS)*. https://www.fiskeridir.no/Tall-og-analyse/AApne-data/elektronisk-rapportering-ers. (Visited on 03/04/2024).

[7] Wes McKinney. "Data Structures for Statistical Computing in Python." In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56 –61. DOI: 10.25080/Majora-92bf1922-00a.

[8] Stefan Behnel et al. "Cython: The Best of Both Worlds." In: *Computing in Science & Engineering* 13.2 (2011), pp. 31–39. DOI: 10.1109/MCSE.2010.118.

[9] The pandas development team. *pandas-dev/pandas: Pandas*. Version v2.2.2. Apr. 2024. DOI: 10.5281/zenodo.10957263. URL: https://doi.org/10.5281/zenodo.10957263.

[10] Filipe et al. *python-visualization/folium: v0.15.1*. Version v0.15.1. Dec. 2023. DOI: 10.5281/zenodo.10255171. URL: https://doi.org/10.5281/zenodo.10255171.

[11] Volodymyr Agafonkin et al. *Leaflet/Leaflet: v1.9.4*. May 2023. DOI: 10.5281/zenodo.7335489. URL: https://doi.org/10.5281/zenodo.7335489.

[12] Pallets. *pallets/flask: Flask*. Version v3.0.0. Sept. 2023. URL: https://github.com/pallets/flask.

[13] www.json.org. *Introducing JSON*. https://www.json.org/json-en.html. (Visited on 02/02/2024).

[14] mozilla.org. *Working with JSON*. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON. (Visited on 02/02/2024).

[15] Python Software Foundation. *JSON encoder and decoder*. https://docs.python.org/3.10/library/json.html. (Visited on 03/07/2024).

[16] Python Software Foundation. *Python object serialization*. https://docs.python.org/3.10/library/pickle.html. (Visited on 03/07/2024).

[17] Marco Slaviero. "Sour Pickles: Shellcoding in Python's serialisation format." In: *Black Hat USA* (2011).

[18] Huanhuan Li, Hang Jiao, and Zaili Yang. "AIS data-driven ship trajectory prediction modelling and analysis based on machine learning and deep learning methods." In: *Transportation Research Part E: Logistics and Transportation Review* 175 (2023), p. 103152.

[19] Huanhuan Li, Hang Jiao, and Zaili Yang. "Ship trajectory prediction based on machine learning and deep learning: A systematic review and methods analysis." In: *Engineering Applications of Artificial Intelligence* 126 (2023), p. 107062.

[20] Duong Nguyen and Ronan Fablet. "A Transformer Network with Sparse Augmented Data Representation and Cross Entropy Loss for AIS-based Vessel Trajectory Prediction." In: *IEEE Access* (2024), 1–1. ISSN: 2169-3536. DOI: 10.1109/access.2024.3349957. URL: http://dx.doi.org/10.1109/ACCESS.2024.3349957.

[21] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

[22] Duong Nguyen et al. "A Multi-Task Deep Learning Architecture for Maritime Surveillance Using AIS Data Streams." In: *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. 2018, pp. 331–340. DOI: 10.1109/DSAA.2018.00044.

[23] Duong Nguyen et al. "GeoTrackNet—A Maritime Anomaly Detector Using Probabilistic Neural Network Representation of AIS Tracks and A Contrario Detection." In: *IEEE Transactions on Intelligent Transportation Systems* 23.6 (June 2022), 5655–5667. ISSN: 1558-0016. DOI: 10.1109/tits.2021.3055614. URL: http://dx.doi.org/10.1109/TITS.2021.3055614.

[24] Gözde Boztepe Karataş, Pinar Karagoz, and Orhan Ayran. "Trajectory pattern extraction and anomaly detection for maritime vessels." In: *Internet of Things* 16 (2021), p. 100436.

[25] Zhiyuan Shi et al. "LSTM-based flight trajectory prediction." In: *2018 International joint conference on neural networks (IJCNN)*. IEEE. 2018, pp. 1–8.

[26] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.

[27] Saeed Arasteh et al. “Fishing vessels activity detection from longitudinal AIS data.” In: *Proceedings of the 28th International conference on advances in geographic information systems*. 2020, pp. 347–356.

[28] Martha Dais Ferreira et al. “A Semi-Supervised Methodology for Fishing Activity Detection Using the Geometry behind the Trajectory of Multiple Vessels.” In: *Sensors* 22.16 (2022). ISSN: 1424-8220. DOI: 10.3390/s22166063. URL: https://www.mdpi.com/1424-8220/22/16/6063.

[29] Erico N. de Souza et al. “Improving Fishing Pattern Detection from Satellite AIS Using Data Mining and Machine Learning.” In: *PLOS ONE* 11.7 (July 2016), pp. 1–20. DOI: 10.1371/journal.pone.0158248. URL: https://doi.org/10.1371/journal.pone.0158248.

[30] C Carl Robusto. “The cosine-haversine formula.” In: *The American Mathematical Monthly* 64.1 (1957), pp. 38–40.

[31] Directorate of Fisheries. *Åpne data: posisjonsrapportering (VMS)*. https://www.fiskeridir.no/Tall-og-analyse/AApne-data/posisjonsrapportering-vms. (Visited on 05/25/2024).

[32] Jennifer L Shepperson et al. “A comparison of VMS and AIS data: the effect of data coverage and vessel position recording frequency on estimates of fishing footprints.” In: *ICES Journal of Marine Science* 75.3 (2018), pp. 988–998.

[33] Pascal Thoya et al. “AIS and VMS ensemble can address data gaps on fisheries for marine spatial planning.” In: *Sustainability* 13.7 (2021), p. 3769.

[34] Directorate of Fisheries. *Posisjonsrapportering (VMS)*. https://www.fiskeridir.no/Yrkesfiske/Rapportering-paa-havet/Posisjonsrapportering. (Visited on 05/25/2024).

[35] Morten Omholt-Jensen. “Vessel destination forecasting based on historical AIS data.” MA thesis. NTNU, 2021.