2022-06

# MARITIME DOMAIN AWARENESS THROUGH THE CHARACTERIZATION OF SHIP BEHAVIOR WITH AIS DATA

Alese, Matthew E.

Monterey, CA; Naval Postgraduate School


# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**MARITIME DOMAIN AWARENESS THROUGH THE CHARACTERIZATION OF SHIP BEHAVIOR WITH AIS DATA**

by

Matthew E. Alese

June 2022

Thesis Advisor: James W. Scrofani
Co-Advisor: Jihane Mimih

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | | *Form Approved OMB No. 0704-0188* |
|---|---|---|---|



| **1. AGENCY USE ONLY** *(Leave blank)* | **2. REPORT DATE** June 2022 | **3. REPORT TYPE AND DATES COVERED** Master's thesis |
|---|---|---|

**4. TITLE AND SUBTITLE**
MARITIME DOMAIN AWARENESS THROUGH THE CHARACTERIZATION OF SHIP BEHAVIOR WITH AIS DATA

**5. FUNDING NUMBERS**

**6. AUTHOR(S)** Matthew E. Alese

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Naval Postgraduate School
Monterey, CA 93943-5000

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
N/A

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**
Approved for public release. Distribution is unlimited.

**12b. DISTRIBUTION CODE**
A

**13. ABSTRACT (maximum 200 words)**

Maritime Domain Awareness (MDA), as defined in the 2005 *National Strategy for Maritime Security,* is the "effective understanding of anything associated with the global maritime domain that could impact the security, safety, economy, or environment of the United States." Thus, it is imperative for the U.S. Navy to develop approaches that enhance understanding of the maritime domain in order to maintain operational effectiveness. One such way to enhance this understanding is to develop approaches that automate the analysis of Automatic Identification System (AIS) data to characterize the behavior of ships in the maritime domain. By the sheer amount of AIS data available, it quickly becomes challenging for a human operator to identify ship behaviors throughout the world. When timeliness is important for decision makers, it becomes even more important that the characterization of ship behavior is done quickly and accurately to identify potential issues or threats. Thus, a major contribution of this thesis is the development of an autonomous machine learning system that characterizes ship behavior quickly and accurately in order to achieve MDA in a particular environment. This includes an autonomous system for the identification of ship tracks in a region. Two major contributions of this work are the development of a taxonomy of ship behaviors, which is currently lacking in the literature, and a report on the characterization of such behaviors through machine learning methods.

**14. SUBJECT TERMS**
maritime domain awareness, MDA, ship behavior, behavior classification, track identification, AIS, machine learning, random forest

**15. NUMBER OF PAGES**
99

**16. PRICE CODE**

| **17. SECURITY CLASSIFICATION OF REPORT** | **18. SECURITY CLASSIFICATION OF THIS PAGE** | **19. SECURITY CLASSIFICATION OF ABSTRACT** | **20. LIMITATION OF ABSTRACT** |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UU |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release. Distribution is unlimited.**

# MARITIME DOMAIN AWARENESS THROUGH THE CHARACTERIZATION OF SHIP BEHAVIOR WITH AIS DATA

Matthew E. Alese
Ensign, United States Navy
BS, United States Naval Academy, 2021

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**
**June 2022**

Approved by: James W. Scrofani
Advisor

Jihane Mimih
Co-Advisor

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Maritime Domain Awareness (MDA), as defined in the 2005 *National Strategy for Maritime Security,* is the "effective understanding of anything associated with the global maritime domain that could impact the security, safety, economy, or environment of the United States." Thus, it is imperative for the U.S. Navy to develop approaches that enhance understanding of the maritime domain in order to maintain operational effectiveness. One such way to enhance this understanding is to develop approaches that automate the analysis of Automatic Identification System (AIS) data to characterize the behavior of ships in the maritime domain. By the sheer amount of AIS data available, it quickly becomes challenging for a human operator to identify ship behaviors throughout the world. When timeliness is important for decision makers, it becomes even more important that the characterization of ship behavior is done quickly and accurately to identify potential issues or threats. Thus, a major contribution of this thesis is the development of an autonomous machine learning system that characterizes ship behavior quickly and accurately in order to achieve MDA in a particular environment. This includes an autonomous system for the identification of ship tracks in a region. Two major contributions of this work are the development of a taxonomy of ship behaviors, which is currently lacking in the literature, and a report on the characterization of such behaviors through machine learning methods.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| AIS | Automatic Identification System |
| COG | course over ground |
| DBSCAN | density-based spatial clustering of applications with noise |
| DKLT | discrete Karhunen-Loeve transform |
| EBSA | ecologically and biologically sensitive area |
| IMO | International Maritime Organization |
| MDA | Maritime Domain Awareness |
| MMSI | maritime mobile service identity |
| MPA | marine protected areas |
| PCA | principal component analysis |
| SOG | speed over ground |
| VHF | very high frequency |
| WCSS | within-cluster sum of squares |

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

First and foremost, I would like to thank God who is the source of all good things and the inspiration for all works. Scientific research is a lens through which we can learn more about His world and I am thankful to be blessed to take a part in learning about His wonderful creation.

I would like to thank my family and friends for their unwavering support throughout this arduous process. I could not have gotten through this and succeeded without your support and care. Thank you for always being there for me.

I would also like to thank Dr. James Scrofani and Dr. Jihane Mimih for kindling my interest in machine learning and being so receptive to taking me on as a student at the beginning when you knew so little about me. Thank you both so much for guiding me throughout this process and providing me your support. I hope that I have lived up to your expectations.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

Maritime Domain Awareness (MDA) is the "effective understanding of anything associated with the global maritime domain that could impact the security, safety, economy, or environment of the United States" [1]. Thus, it is imperative the U.S. Navy develop MDA in order to maintain operational effectiveness. One such way to develop MDA is to analyze Automatic Identification System (AIS) data to characterize the behavior of ships in the maritime domain.

AIS is a very high frequency (VHF) broadcast system that provides information about ship static and dynamic information. Static information includes details such as its Maritime Mobile Service Identity (MMSI), ship type, ship name, and intended destination. Dynamic information includes details such as its position, course over ground, heading, rate of turn, navigation status, speed over ground, and time of measurements. Additionally, the International Maritime Organization (IMO) requires "AIS to be fitted aboard all ships of 300 gross tonnage and upwards engaged on international voyages, cargo ships of 500 gross tonnage and upwards not engaged on international voyages and all passenger ships irrespective of size" [2]. However, many ships carry AIS for both safety and navigational enhancement. Therefore, AIS provides a vast amount of information from which much about the maritime domain can be inferred through machine learning methods.

Rutherford D. Roger has stated: "We are drowning in information and starving for knowledge" [3]. In the age of the internet, this could not be truer with the emergence of the "big data" phenomenon. As defined by the McKinsey Global Institute in May 2011: "Big data refers to data sets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze" [4]. Indeed, every one minute of the day—Google conducts 5.7 million searches, Twitter users post 575,000 tweets, and Amazon customers spend $283,000 [5]. Spire, a company that specializes in providing AIS data to commercial consumers, generates nearly 180 million AIS records a day for its full service [6]. Such massive amounts of data make analysis using conventional tools difficult for operators who cannot process nor fully exploit the information contained in the collected data. We therefore live in the age of big data and the only way out is through automation and machine learning. Richard Hamming,

a distinguished NPS Computer Science professor, remarked that "the purpose of computing is insight, not numbers" [7]. Thus, the overarching goal of this thesis is to provide additional insight towards an understanding of the maritime domain through machine learning.

## A. THESIS OBJECTIVES

The first objective of this thesis is to improve understanding of the maritime domain by developing an approach to automate the analysis of ship behaviors using AIS data. Ship behavior is highly dependent upon the type of ship. Certainly, a warship will be engaged in different activities and have different capabilities than a cargo ship, tanker, or fishing vessel. Some potential ship behaviors may include refueling, tracking, transiting, stopping, and fishing. One major contribution of this thesis will be to develop a taxonomy of ship behaviors that is currently lacking in the literature and that will be useful in development of novel machine learning approaches to characterize ship behavior.

As the world has become increasingly more connected with globalization, so too have the world supply chains. Thus, any disruption in the global supply chains can have massive negative impacts on the global economy. A look at some recent current events confirms the devastating impact that disruptions in the global supply chain can cause to the global economy. In March 2021, a cargo ship, the *Ever Given,* was lodged sideways in the Suez Canal and caused a massive rerouting of shipping—freezing nearly $10 billion in trade per day [8]. Additionally, in November 2021 many cargo ships were left waiting to unload outside of the Los Angeles harbor, where nearly 40% of imports enter the U.S., as supply-chain congestion caused the waiting time to nearly double [9]. Thus, the emergence of new routes could also aid in the identification of global supply chain issues that could save both time and resources. Therefore, a second contribution of this thesis consists of the development of a system for the automatic identification of tracks.

By the sheer amount of AIS data available, it quickly becomes difficult for a human operator or analyst to identify ship behaviors. When timeliness is important for decision makers, it becomes even more important that the characterization of ship behavior be done quickly and accurately to identify potential issues or threats. Machine learning is therefore one method to enable automation for solving this problem. Thus, a third contribution of this

thesis is to develop a machine learning system to characterize ship behaviors and/or anomalies quickly and accurately in order to achieve MDA in a particular environment.

## B. RELATED WORKS

Past methods in the literature have used machine learning methods on AIS data to model and solve a variety of tasks including, but not limited to, AIS spoofing detection [10], port destination prediction [11], and ship trajectory prediction [12]. This thesis leverages the results and methods of these researchers to expand and focus upon the broader area of ship behavior detection and classification.

In the area of ship behavior classification, some of the current work has focused on fishing classification. Particularly, [13] and [14] both work on the classification of different types of fishing activities. Fishing has often been the focus of previous work on ship behavior classification due to the "monitoring of marine fisheries and the enforcement of spatial management measures, such as marine protected areas (MPAs), ecologically and biologically sensitive areas (EBSAs) as well as fisheries closure zones" [14]. Therefore, this thesis seeks to expand this work by including other behaviors in the classification process.

## C. THESIS ORGANIZATION

This thesis is organized into six chapters. Chapter II discusses background information on AIS data, the Hough transform, several clustering algorithms (k-means and DBSCAN), several classification models (decision tree and random forest), and the use of features and principal component analysis for machine learning. Chapter III then builds a comprehensive ship behavior and ship type taxonomy that will help inform the algorithms used throughout this thesis. Chapter IV then details some data pre-processing steps taken to improve performance of the algorithms contained within this thesis. This includes geographic filtering, geographic sectoring, downsampling, feature extraction methods and how principal component analysis was utilized in this thesis. Chapter V describes the results of the algorithms used for both automatic track identification and the ship behavior classifiers. Finally, Chapter VI outlines key conclusions from the work contained within this thesis and offers several recommendations for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

# II. BACKGROUND

This chapter is organized into five different sections. The first section details the organizational structure of AIS data and related data formats for future related works. The second section explains the Hough Transform as used to generate ship track information. The third section explains various clustering algorithms, in particular the DBSCAN algorithm, which will be used in tandem with the Hough Transform to automatically identify tracks. The fourth section includes various machine learning classifiers that will be utilized for ship behavior classification. Finally, the fifth section describes the use of features in machine learning and how principal component analysis can be formulated to potentially boost classification performance.

## A. AUTOMATIC IDENTIFICATION SYSTEM

AIS promotes safety and collision avoidance at sea by sharing ship data with other nearby vessels along with maritime authorities. AIS was developed in the early 2000s in order to implement goals of "automating and improving navigation safety at sea, preventing collisions through a ship-to-ship operative mode, providing key information about a ship and its cargo to other ships and parties within littoral zones, and functioning as a traffic management tool for Vessel Traffic Services" [15]. Additionally, AIS provides more than just navigational data. It also relates information specific to the vessel itself as well as information pertaining to a voyage. A summary listing of the data given by the AIS [15] is shown in Table 1.

Table 1. AIS data fields. Adapted from [15].

| **Static Information** | **Navigational Information** | **Voyage-specific Information** |
|---|---|---|
| Maritime Mobile Service Identity (MMSI) | Course over Ground (COG) | Ship Draught |
| International Maritime Organization (IMO) Number | Speed over Ground (SOG) | Hazardous Cargo |
| Call Sign | Heading | Destination and ETA |
| Ship Name | Navigational Status | Route Plan |

| Static Information | Navigational Information | Voyage-specific Information |
|---|---|---|
| Length and Beam | Rate of Turn | |
| Type of Ship | Position (Latitude/ Longitude) | |
| Location of Position Fixing Antenna on Ship | | |

AIS provides a wealth of information on ships that can then be utilized to learn meaningful information. Many types of ships are required to have AIS, and the total amount of data available is staggering. In fact, Spire (a company specializing in providing satellite AIS data to end users) acquires approximately 180 million AIS records per day [6]. Therefore, it is necessary that an automated machine learning approach is used to help enable the operator or decision maker to make timely and necessary decisions.

However, AIS indeed has some vulnerabilities. While transmission is often required by the IMO, AIS may not be transmitting either due to faulty equipment or operator intervention. Additionally, since static and voyage information are manually entered by the crew, the information contained in those fields may be empty or wrong. Whether done intentionally or not, this can be a potential security issue for nearby ships. Thus, any method employed using AIS data requires a cooperative actor. However, the methods used on AIS data can indeed be supplemented with data from other sources—a process called "multi-int signal processing," which ensures a "fuller, more insightful operating picture than processing intelligence data from a single source alone can create" [16]. Regardless, using AIS data alone on cooperative vessels can help build a picture of representative behavior of ships to be used in the future on uncooperative ships.

## B. HOUGH TRANSFORM

The Hough transform is often used in line detection methodologies, particularly in the field of image processing [17]. Since ship navigators often find it easiest to travel in straight lines on constant bearings, the Hough transform will be applied to detect straight segments of a ship track. First, to understand this transformation, the parametric representation of lines must be discussed.

## 1. Parametric Representation of Lines in Geometry

A straight line in the *x*-*y* cartesian plane is familiarly described by two parameters: its slope ($m_0$) and *y*-intercept ($b_0$). The usual parametrization is shown as:

$$y = m_0 x + b_0 \tag{1}$$

However, computationally this representation can cause issues if the line is vertical, which may happen in this applied case if a ship is travelling due north/south, as the slope will go to infinity or negative infinity. Thus, a more useful representation of lines is with *r* and $\theta$ parameters as outlined in "Use of the Hough Transformation to Detect Lines and Curves in Pictures" [17]. The $\theta$ parameter is the angle to the normal of the line from the origin and the *r* parameter is the distance from the origin to the normal of the line. If $\theta$ is restricted to (-180, 180°) and *r* is restricted from [0, ∞) then every straight line has a unique set of parameters that describe it. This "normal parameterization" of straight lines is shown in Figure 1.

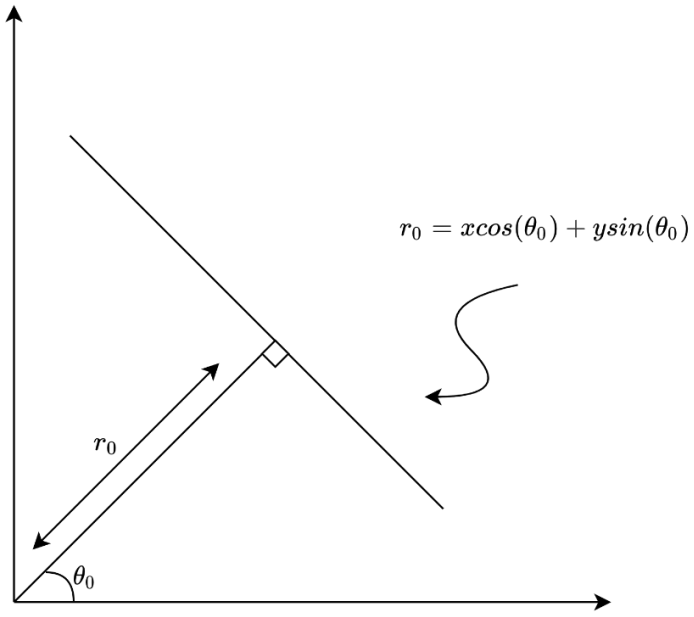



Figure 1. Normal parameterization of a line. Adapted from [18].

Thus, each line can be represented using two parameters, $r$ and $\theta$. Using these parameters, the line equation of infinite points in (1) can be rewritten using Duda and Hart's normal parameterization [17] as:

$$r_0 = x\cos(\theta_0) + y\sin(\theta_0) \tag{2}$$

## 2. Representation of a Single Point in Parameter Space

From Equation (2), it was shown that a line of infinitely many points can be represented by two parameters, $r$ and $\theta$. Similarly, it may be seen that a single point $(x_0, y_0)$ in the $x$-$y$ plane has infinitely many lines running through it. Therefore, a point in the $x$-$y$ plane corresponds to a sinusoidal curve in the $\theta$-$r$ plane [18]. A summary of point-curve transformations (Hough transform) is shown in Figure 2. As demonstrated in Figure 2, colinear points in the $x$-$y$ plane create a corresponding number of sinusoidal curves which intersect at a single point in the $\theta$-$r$ plane. The representation of all these lines is described by a sinusoidal curve in the parameter plane as:

$$r = x_0\cos(\theta) + y_0\sin(\theta) \tag{3}$$

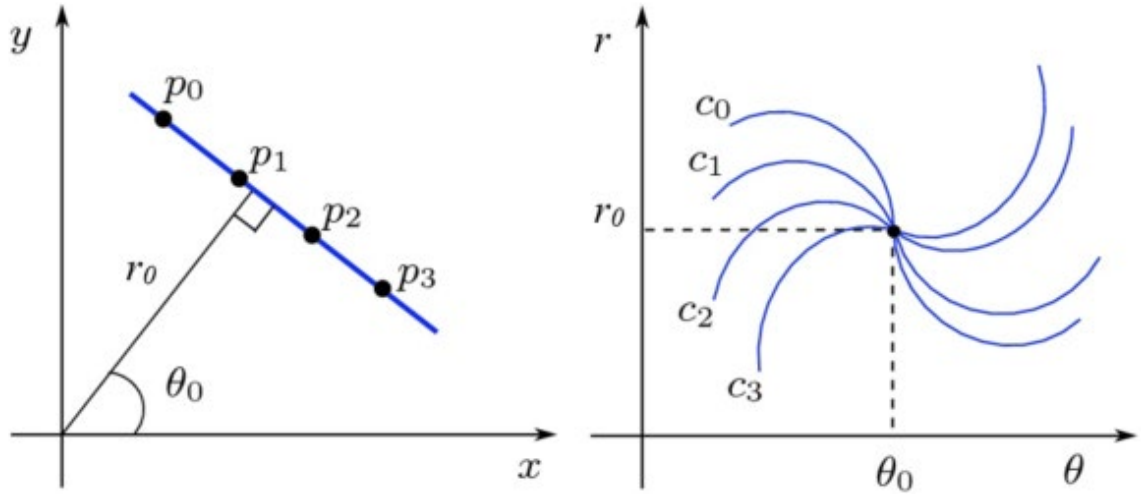



Figure 2. Point-curve Hough transformations. Adapted from [19].

Thus, the Hough Transform between points in $x$-$y$ coordinate plane to the $r$-$\theta$ parameter plane can be used to detect colinear points by finding the intersection of $r$-$\theta$

curves. However, with the sheer amount of data that AIS provides, each positional update of a ship requires a *r-θ* curve to be computed. If this colinear detection is to be automated, then each curve requires many points to approximate it, which may be computationally intensive. To reduce the computational load for line detection, a second method is proposed to reduce the computational load, thus making it more feasible in the presence of large amounts of AIS data.

Similarly, if there are a finite number of points in the *x-y* plane then each pair of points in the *x-y* plane forms a line and a set of corresponding points in the *r-θ* plane. If there are *n* distinct points in the *x-y* plane, then the total number of pairings and thus *r-θ* parameters calculated is represented as a combination of the points as shown in Equation (4).

$$S = {}_nC_2 = \frac{n!}{2!(n-2)!} = \frac{n(n-1)}{2} \tag{4}$$

This means that *n x-y* coordinate plane points are transformed into a set S of size *n(n-1)/2* points. Alternatively, each individual point would be represented by an infinite number of points. Next, it is observed that point pairings that are colinear to other point pairings will have normal parameters that are identical. If they are not on the same line, then the pairings should be spaced farther apart in the *r-θ* plane. This transformation between pairings and parameters is shown in Figure 3.

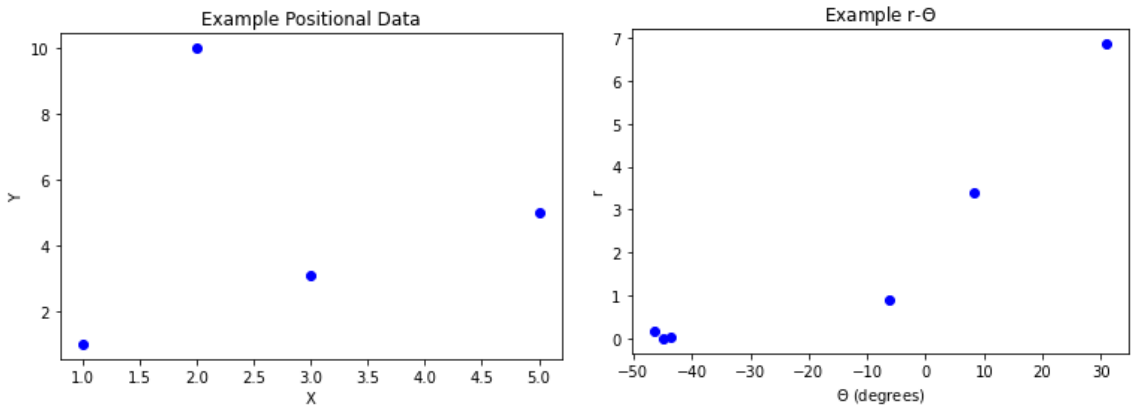



Figure 3. Point pairings to r-theta space transformation. A cluster of $r$-$\theta$ points corresponding to the line present in the positional data is located near θ = -45° and $r$ = 0.

This idea of using closeness to distinguish points from one another is known as clustering in a machine learning context. Thus, colinear points can be identified by identifying clusters of parameter points. Thus, similarly ship positional data can be used to identify linear tracks through the Hough transform.

## C. TRACK IDENTIFICATION USING CLUSTERING ANALYSIS

Clustering analysis seeks to group "data objects based only on information found in the data that describes the objects and their relationships" [20]. In this instance, clustering analysis will be applied to the data after the Hough transform is applied as discussed in Section II.B. Thus, the $r$ and $\theta$ values will be clustered to find unique groups. Since the clustering analysis is being done to determine meaningful associations without knowing which groups the points belong to beforehand, cluster analysis is often known as "unsupervised learning" [20]. The goal of this clustering analysis is to determine meaningful clusters of the $r$ and $\theta$ values that will determine unique lines that the original data points lie upon. The following discussion of the method used for track identification will take the trajectory of a ship in the Baltic Sea as an example. The ship trajectory—derived from real AIS data—is shown in Figure 4.

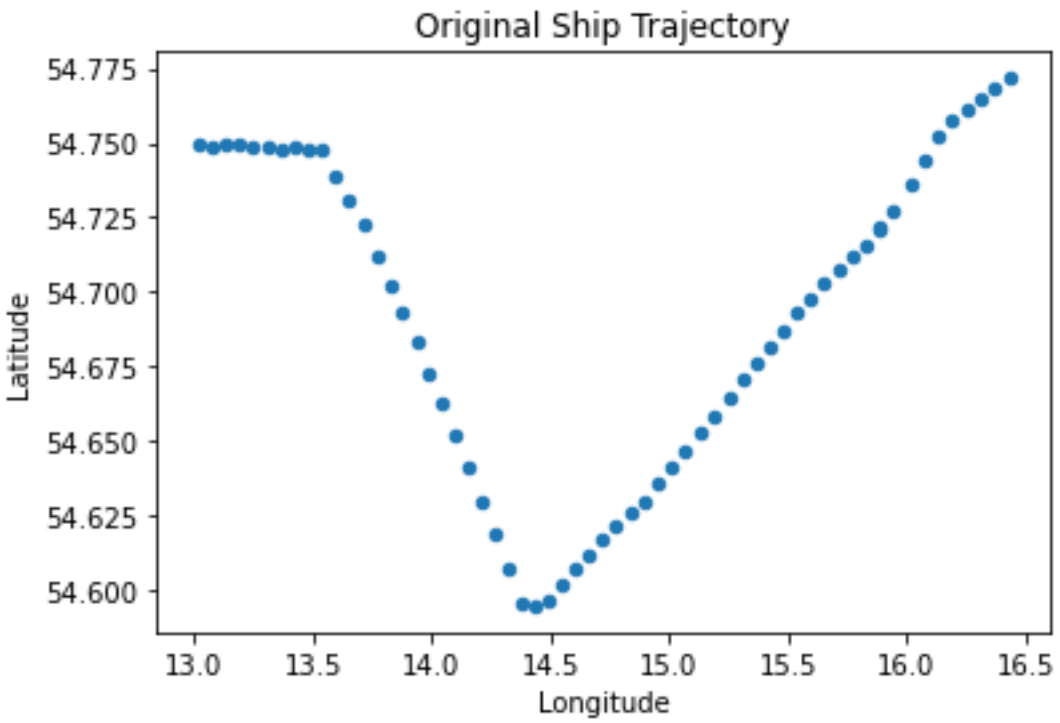



Figure 4. Ship trajectory as it sails through the Baltic Sea.

From Figure 4 it appears that there are three distinct straight trajectories that the ship is sailing upon. As a result, it is expected that the clustering analysis should yield three sets of parameters that describe each of these lines.

### 1. Feature Standardization

Often machine learning algorithms, including those used for clustering, are sensitive to the scale or size of the individual feature values. Clustering analysis, in particular, is sensitive to the scale of individual features [21]. A feature is a meaningful attribute about a data object: in this case the ship latitude and longitude are both features. Thus, before analysis is conducted it is often important to rescale the dataset. One way to standardize a dataset is by transforming each feature so that it is of zero mean and unit variance. In order to rescale the features, the following transformation is applied:

$$y_i = \frac{x_i - \bar{X}}{s} \tag{5}$$

For this equation, $x_i$ is the original data point, $\bar{X}$ is the sample mean, $s$ is the sample standard deviation, and $y_i$ is the resulting transformed variable. Thus, the original ship trajectory was scaled using the feature standardization transformation as defined in Equation (5). The scaled ship trajectory is shown in Figure 5.

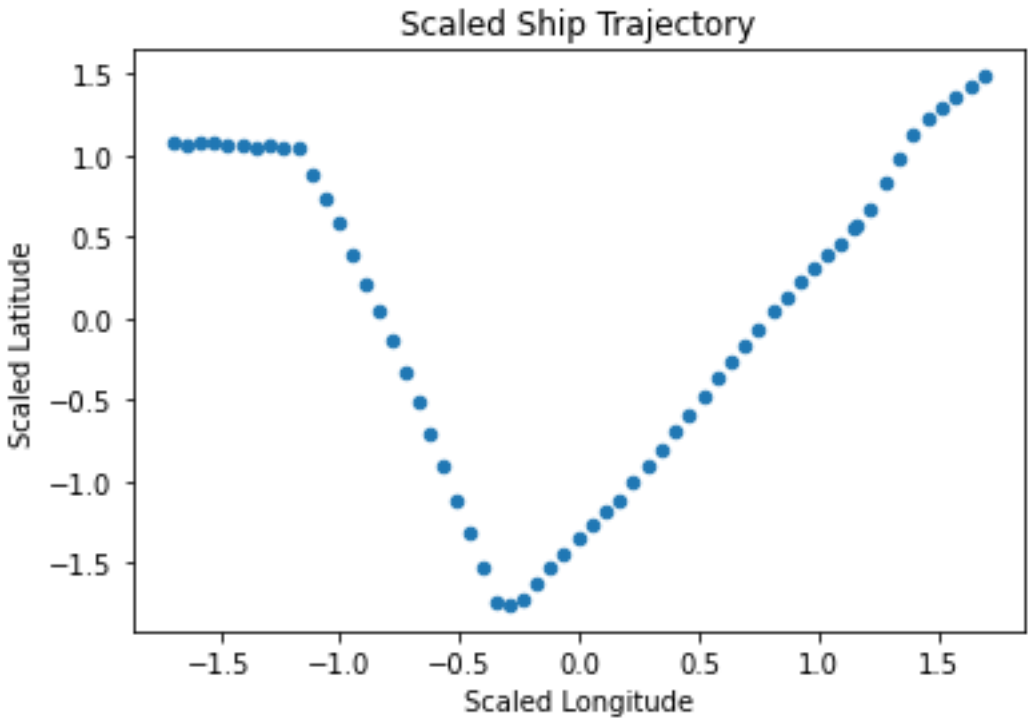



Figure 5. Scaled ship trajectory as it sails through the Baltic Sea.

After the ship trajectory was scaled, the data points were then used to compute $r$ and $\theta$ parameters as discussed in Section II.B. Since there were 61 instances of each feature, there were 1830 sets of $r$-$\theta$ parameters computed as shown by Equation (4). The result of the Hough transform on the ship trajectory depicted in Figure 5 is shown in Figure 6.


Figure 6. Hough transform as applied to the scaled ship trajectory data.

Thus, as expected, there indeed appear to be three areas of higher density that can be clustered in order to find the straight lines through which the ship is navigating. However, it also appears that there is a significant number of noise points that do not correspond to any of the three straight lines. Any method used to cluster should have a way to deal with these noise points so that they do not bias or distort the estimated parameter values for the lines through the straight portions of the track.

### 2. K-means Clustering Algorithm

The first, and often simplest, clustering algorithm available for processing is known as the k-means clustering algorithm. The k-means clustering algorithm works by partitioning the dataset into $k$ distinct groups or clusters. First, $k$ initial centroids are chosen where $k$ is a user-specified parameter relating to the number of clusters desired to make from the data. Then each data point is assigned to the closest centroid as being a member of that cluster. The distance to the closest centroid can be defined by a variety of measures. However, the most common distance measure used to determine proximity is the Euclidian

distance. Next, the centroid is updated based upon the new information, i.e., the centroids are recomputed from their new datapoints that are included within the cluster. This process of reassigning the datapoints to the closest centroid is done iteratively until the centroids each converge to a constant value. A pseudocode description of the k-means clustering algorithm is shown in Table 2.

Table 2. k-means Clustering Algorithm. Source: [20].

| |
|---|
| 1: Determine *k* initial centroids |
| 2: While centroids do not converge |
| 3: Assign each data point to a cluster according to its closest centroid by Euclidean distance |
| 4: Recalculate the centroid of each cluster based upon the location of the newly assigned data points |

However, as discussed previously, the only user-defined parameter for the k-means clustering algorithm is "*k*," the number of clusters. Thus, it is necessary to have a priori knowledge of the dataset to correctly determine how many clusters should be expected. An alternative solution is to solve the k-means clustering problem iteratively with multiple values of *k* and determine an optimal *k* at which the error greatly decreases. One such error measure is known as the within-cluster sum of squares (WCSS) [22]. The WCSS is computed by taking the Euclidian distance from each data point to its assigned centroid, taking the square value, and then adding all such errors together. Thus, if the data points are far away from their centroid, the WCSS will be high. Theoretically, if all data points were exactly located on their centroid, then the error would be zero. Thus, the WCSS is a viable error or distance measure that can be used to optimize the algorithm.

Once way to choose the number of clusters for analysis is by determining the WCSS iteratively for multiple values of *k*. One popular method for choosing *k* is known as the "elbow" method [22]. A WCSS versus *k* plot for the ship sailing through the Baltic Sea is shown in Figure 7.

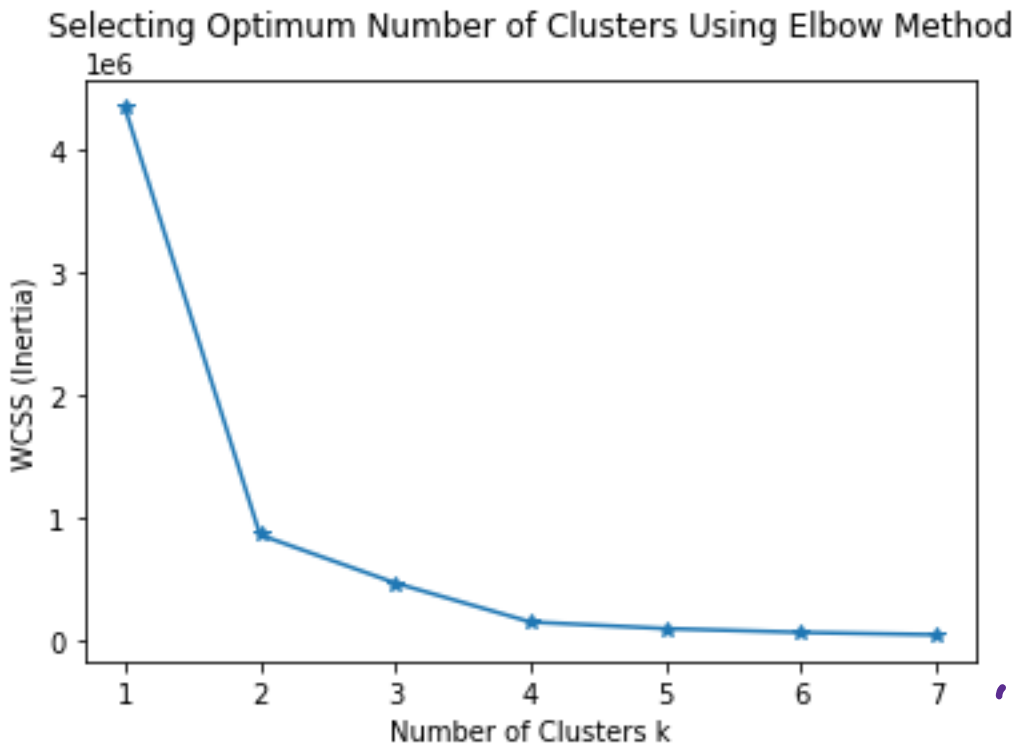



Figure 7. Determining the optimal number of clusters using the WCSS as an error measure.

For the "elbow" method, the number of clusters is chosen when the WCSS error score stops decreasing at a rapid rate. Based upon the results in Figure 7, it appears that this drop-off occurs around $k = 3$ or $k = 4$. Thus, as expected, the number of clusters chosen that significantly decreases the error is three, which corresponds with the number of straight lines in the data.

However, another drawback to the k-means algorithm is that it includes all data points in the computation of the centroids. It does not account for points that may simply be noise in the data—a circumstance that is common in measurements. As a result, the noise may significantly impact the estimation of the $r$ and $\theta$ parameters, which characterize the lines. Instead, it is desirable for these noise points to be disregarded to obtain a better estimation of the optimal $r$ and $\theta$ track parameters.

As discussed previously, a major goal of this thesis is to develop a method for analysis that can be implemented in real-time applications to help automate tasks for the

operator or decision maker. Thus, since every situation will be different and often unknown, the number of clusters for the k-means clustering algorithm will need to be determined at each application of the algorithm. This coupled with the fact that the k-means algorithm includes all data points in its analysis, especially those that can clearly be characterized as noise, means that the k-means algorithm is not particularly suitable for the real-time analysis of ship tracks for MDA. Therefore, the k-means clustering algorithm was rejected from consideration for use in this thesis.

### 3. Density-Based Spatial Clustering of Applications with Noise

Another clustering algorithm that has been developed is known as the density-based spatial clustering of applications with noise (DBSCAN). DBSCAN works by considering the density of points. In contrast to k-means, DBSCAN automatically determines the number of clusters and also identifies points in low-density regions as noise and removes them [20]. Thus, the two main disadvantages of the k-means clustering algorithm are immediately addressed by DBSCAN.

The DBSCAN algorithm classifies every data point as one of three types—a core point, border point, or noise point. A core point is on the interior of a cluster and is the basis for defining a cluster. A data point is classified as a core point if the number of surrounding points within a distance measure threshold—defined as *Eps*—exceeds a "number of points" threshold—defined as *MinPts*. Both *Eps* and *MinPts* are user-defined parameters for the model that enhance the selectivity of the model. Thus, the core points define a cluster. A border point does not meet the requirements of a core point but falls within the Eps radius of a core point [20]. Thus, a border point is included in the cluster defined by the surrounding core points. Lastly, a noise point is any data point that does not meet the requirements of either a core point or a border point. Thus, the DBSCAN algorithm automatically determines the number of clusters and eliminates noise points by classifying all data points into each of the three categories.

Thus, the main advantage of the DBSCAN algorithm is that it automatically determines the number of clusters and removes noise points. However, the main disadvantage of the algorithm is selecting optimal values for the *Eps* and *MinPts*

parameters, which are often tuned iteratively or with some knowledge of the kinds of values the dataset is expected to have.

For the case examined previously, the $r$ and $\theta$ parameters for the ship track were clustered using the DBSCAN algorithm as shown in Figure 8. The parameters chosen for this algorithm are *Eps = 0.2* and *MinPts = 10*.

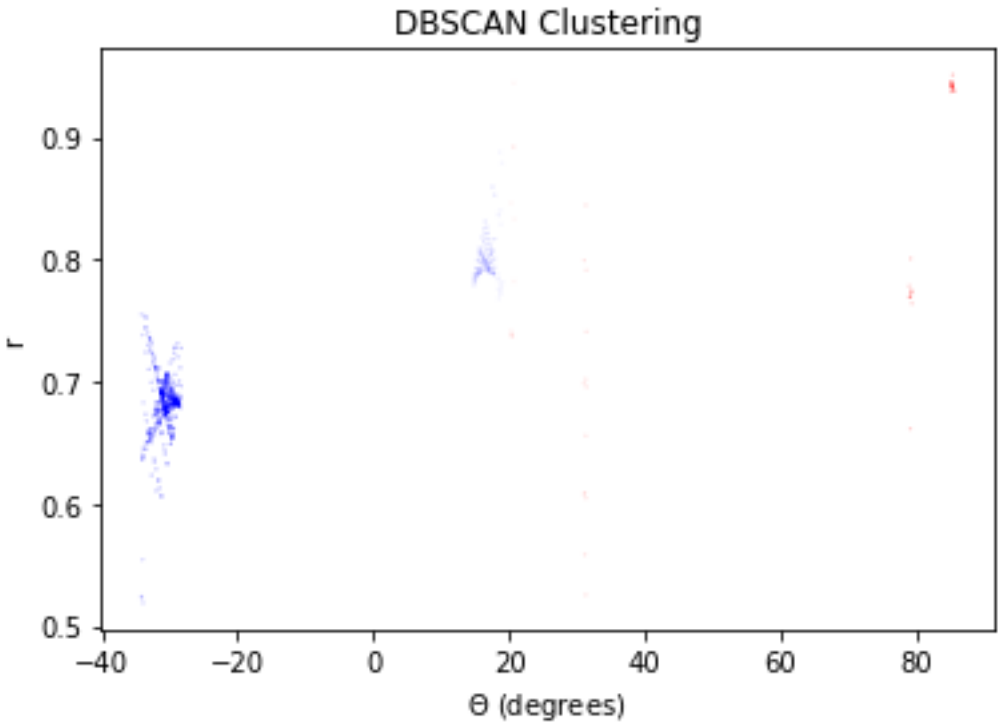



Figure 8. Determining the optimal number of clusters using the WCSS as an error measure.

Thus, the results from the DBSCAN clustering show that the algorithm successfully removed many of the points that would have been considered noise. After clustering, there appears to be three higher density clusters as expected since there are three straight tracks in the ship trajectory. From this, the $r$ and $\theta$ parameters can be extracted by taking the centroid of the most populous cluster, which appears to be the cluster located near $\theta_0$ = -35° and $r_0$ = 0.65. Thus, the line with the most points has been identified. The line fitting the line on the original track data is shown in Figure 9.

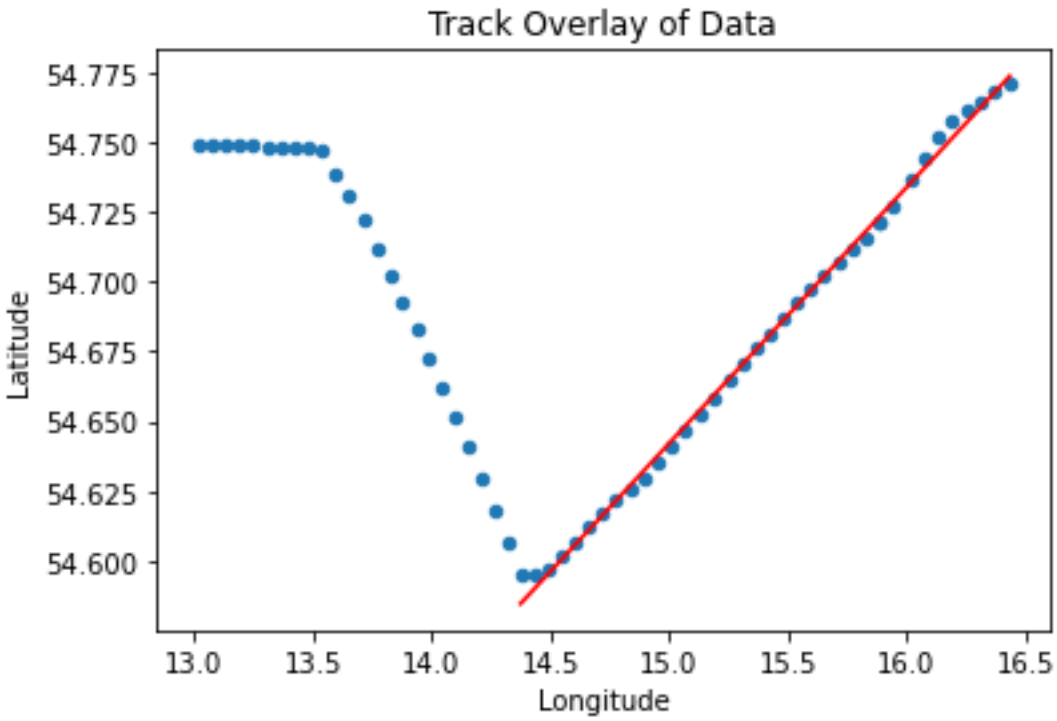



Figure 9. Line fit to original track data

Since the DBSCAN algorithm was successful in identifying the most populous straight-line tracks, it was selected as the clustering algorithm for use throughout this thesis.

It is important to note that variations of the Kalman filter are traditionally used to track ship vessels—both for the prediction of future tracks and for the optimal estimation of their track [23]. However, this method of track estimation was not used for several reasons. The Kalman filter requires track data from individual ships. However, the goal of this analysis is to characterize the most popular tracks in a region and not to characterize the tracks of individual ships. Therefore, batch processing is both necessary and desired. The technique chosen for this analysis (Hough transform and DBSCAN clustering) allows greater data flexibility and for batch processing to occur.

## D. CLASSIFICATION MODELS

Another goal of this thesis is to be able to classify and interpret ship behavior from its spatiotemporal data contained within the AIS data. In order to accomplish this, a classification model must be developed. Classification is the task of mapping data object

features to a class label. Mathematically, classification can be viewed as a mapping of features ($x$) to one of n classes ($C$) as shown in Equation (6).

$$f(x_k) \rightarrow \widehat{y_k} \in \{C_1, C_2, ..., C_n\} \tag{6}$$

In Equation (6), $k$ refers to the *kth* data object in a dataset. Therefore, $x_k$ is the set of features for the *kth* data object and $\hat{y}_k$ is the class estimated from the set of features for the *kth* data object using the classification model $f(\cdot)$. In this way, a classification model maps a set of features to a corresponding set of classes.

As discussed previously, a goal of this thesis is to develop MDA through the classification of ship behaviors. Therefore, the models that are used to classify those ship behaviors must have great interpretability while retaining performance. Two types of models that have great interpretability are decision tree models and random forest models.

### 1. Classification Model Performance Metrics—Computation and Visualization

Before discussing specific classification schemes, it is important to develop an intuitive understanding of classifier performance, especially with regards to an unevenly distributed dataset. First, several terms must be defined before deriving several metrics. A true positive (TP) is a data object that belongs to the class of interest and has been predicted correctly by the classifier. A true negative (TN) is a data object that does not belong to the class of interest and has been predicted correctly by the classifier. A false positive (FP) is a data object that has been misidentified by the classifier as belonging to the class of interest when it does not. A false negative (FN) is a data object that has been misidentified by the classifier as not belonging to the class of interest when it truly does. These definitions hold whether talking about a binary classifier (a classifier that is deciding between two classes) or a multi-class classifier (a classifier that is deciding between more than two classes). All performance metrics of a classifier are then based upon these four outcomes of the classifier (TP, TN, FP, FN).

The first, and perhaps most intuitive performance metric is accuracy. Accuracy is defined in [21] as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Thus, accuracy is a measure of the correctness of a classifier out of all its predictions. Currently, it may now be questioned why go further than this? Can there be other performance metrics better than this? After all, should we not just be attempting to determine the correctness of our classifier? Accuracy as a performance measure, however, can be misleading with an unbalanced dataset, which is a dataset that has more of one class than the others. For example, if a model were attempting to classify if someone has cancer, it is possible that the training data would be perhaps 95% non-cancer patients and 5% cancer patients. A model that only guesses that a person does not have cancer (and therefore cannot ever find a person who has cancer) will indeed be 95% accurate. However, it is very unlikely the classifier will ever find the patient with cancer. This is an example of how accuracy can be misleading and a poor performance metric under certain circumstances. This motivates the formation of three other performance metrics: precision, recall, and F1-score. Precision is defined in [21] as:

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

Thus, precision is a ratio between its true positives over all positive predications. Intuitively, precision is therefore the probability that if a model predicts that an object belongs to a class of interest the model is correct. Another performance metric is recall [21]:

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

Thus, recall is a ratio between its true positives over all actual positives. Intuitively, recall is the probability that if an object belongs to a class of interest, the model will predict it correctly. However, typically, it is desirable to achieve a balance between recall and precision [21]. The F1-score defined in Equation (10) seeks to quantify the balance between the two.

$$F1 = 2\frac{Precision * Recall}{Precision + Recall} \tag{10}$$

The F1-score then is a measure of combined precision-recall performance that is sensitive particularly to low scores. These four performance metrics are used to evaluate the performance of machine learning models used in this thesis.

A multi-class classifier, i.e., a classifier that seeks to classify data into more than two classes are presented near the end of this thesis for the classification of ship behavior. In the case of this thesis, it is desirable to be able to visually identify misclassifications. The confusion matrix indeed was designed for this purpose [24]. A confusion matrix is a summary of classifier correct and incorrect predictions given in both number and in class type. An example of a binary classifier confusion matrix setup is shown in Figure 10.

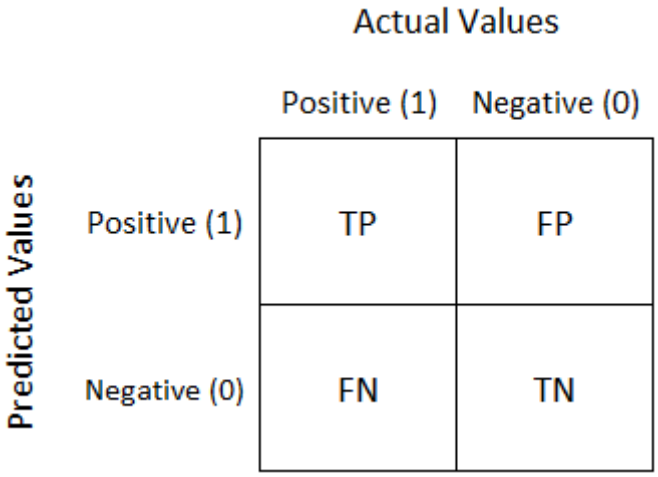



Figure 10. Confusion matrix for a binary classifier. Source: [24].

The performance metrics defined in this section are explanatory for a binary classifier but sometimes it is necessary to visualize the model, particularly for complex multi-class classifiers. A classifier with perfect classifications, where the predicted values match actual values, would thus be diagonal (to borrow terminology from linear algebra). Additionally, a confusion matrix is particularly helpful in multi-class tasks as they make is possible to visualize which class or classes are getting more easily "confused." If two

classes are commonly mistaken for one another then this is suggestive that the classes themselves are closely related to one another.

### 2. Decision Tree Model

A decision tree model is a highly interpretable and visualizable model that "is a hierarchical structure consisting of nodes and directed edges" [20]. In a decision tree, nodes are traveled using directed edges after evaluating features to previously established criteria. A tree has three types of nodes, namely root nodes, internal nodes, and leaf nodes [20]. A root node notes the beginning of a decision tree model and has zero or more outgoing edges. Internal nodes are the nodes which are between root nodes and leaf nodes. Therefore, they have one incoming edge and two or more outgoing edges. Leaf nodes have one incoming edge and note the end decision of a decision tree [20]. An example of a decision tree in the classification task of grasshoppers vs. katydids is shown in Figure 11.

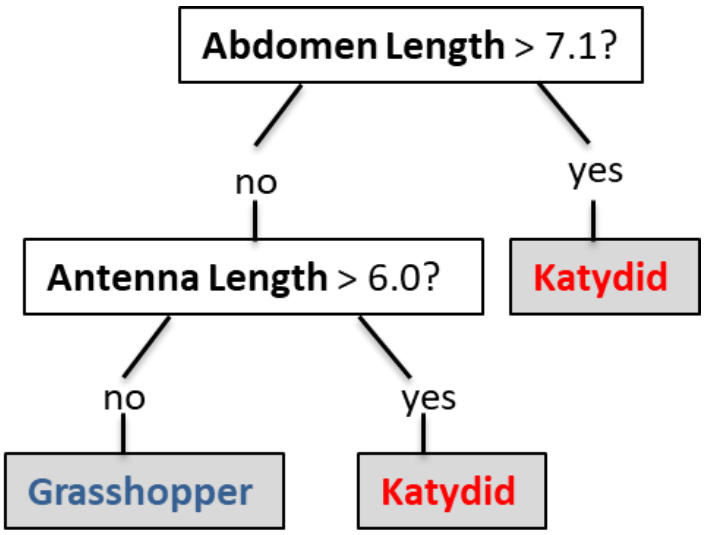



Figure 11. Example of decision tree classification of grasshoppers and katydids. Note that each non-leaf node includes a decision to be made about a feature about the specific insect. Source: [25].

In this example, the root node is the node at the top of the hierarchy. An internal node is the node with the decision about the antenna length. Lastly, the leaf nodes contain

the actual classification decision. Thus, decision tree models are easily able to be visualized and have great interpretability, especially for data with low numbers of features.

However, since there can exist an exponentially high number of decision trees possible for a dataset with many different features, there needs to be a method to produce reasonably accurate, although suboptimal decision trees, in a fast enough amount of time. One such algorithm for growing decision trees is the CART algorithm. The CART algorithm is a type of greedy algorithm—an algorithm that does not find the optimal solution but rather a solution with locally optimal sub-solutions [26]. The CART algorithm was used for the models contained within this thesis through the scikit-learn library in Python [27].

Before explaining the CART algorithm, a measure to define how well a node splits the data needs to be defined. The split measure must be defined in terms of the class distribution of the dataset before and after each split. A split that results in a purer partition of classes will have a lower impurity at the decision node—and thus is a better split. One popular impurity index is called the Gini index [20]. The Gini impurity at node $t$ in a classification task with $n$ classes is defined mathematically in Equation (11).

$$Gini(t) = 1 - \sum_{i=1}^{n} p(i|t)^2 \tag{11}$$

In the Gini impurity measure, $p(i/t)$ refers to the fraction of the data objects that belong to the *ith* class at node $t$. In a binary classification if the split is (0,1) then the Gini impurity is at a minimum of zero. If the split is (0.5,0.5) then the Gini impurity is at a maximum of 0.5. Thus, the decision tree needs to be constructed to minimize the impurity at each individual node. When all node impurities are thus minimized, the model performance should be maximized. Therefore, the decision tree is grown by minimizing the Gini impurity at each split. However, in order to converge to a final result, a stopping criterion must be defined. One way to define a stopping criterion is by testing whether all data objects on a leaf are within the same class label or if this impurity has fallen below a certain threshold.

Now, the decision tree can be determined from the given training data using the CART algorithm with knowledge of the Gini impurity index and the stopping criteria [28]. Thus, for each feature you find the best split. For a feature with *N* different values, there exists *N-1* splits. After each split, the Gini index is calculated. The split that minimizes the impurity is the split that is kept. If a given node satisfies the stopping criterion, then that node is terminated and termed a leaf node. Once the stopping criterion has been reached for all nodes, this signifies that all of the nodes are leaf nodes, and the final decision tree has been grown.

When a decision tree has been fully formed and optimized—it is possible that the tree is too well developed and too large. This may in turn cause the tree to overperform on training data but underperform on testing data—a phenomenon known as overfitting [29]. At some point, a model may become so complex that it is able to perfectly represent the training data—including the variations due to the noise. Thus, the model learns the particulars of the training data while missing more general trends. When a model overfits on training data, the error on the testing data will increase. There are generally two main methods to deal with this phenomenon—reduce the model complexity, a technique aptly called pruning for decision trees, or introduce randomness to the model with variations.

### 3. Random Forest Model

The random forest model is an extension of the decision tree model and was first suggested by Breiman in 2001 [30]. The random forest model is an example of ensemble learning—a method in which multiple models are trained then combined to solve a task (in this case classification). The random forest model therefore is made up of many decision trees that are randomly generated. After many trees are generated, often over 50–100, the consensus of decision tree outputs is used to make a final determination of class. Therefore, the outcome of the random forest is the class that has the consensus of the decision tree outcomes.

A very simple way to grow a random forest is by "selecting at random, at each node, a small group of input variables to split on" [30]. Therefore, this introduces randomness to each of the individual decision trees. Then each decision tree is grown using

a decision tree growth method like the CART methodology described in Section II.D.2. Using this method, the desired number of decision trees can then be grown for the random forest. After all the decision trees are grown, the random forest model has also been successfully grown. The results shown in [30] also demonstrate that the random forest model outperforms the singular decision tree rather decisively on 19 historical datasets typically used for model benchmarking.

A major advantage of the random forest model is that they "do not overfit as more trees are added, but rather produce a limiting value of the generalization error" [30]. Therefore, random forests solve the primary issue of decision trees—overfitting—by introducing randomness into the model in the form of random feature splitting. Indeed, the random forest model is a probabilistic model that achieves a limiting performance as the number of trees increases.

Another advantage of the random forest model is that the relative importance of features can be determined from the model. Thus, the most important features for classification can be identified by computing the impact that the splitting of the feature has on decreasing the impurity within each tree. The mean of these impurity differentials can then be taken and visualized in order to determine the most important features for each random forest model classification [31]. For this thesis, which seeks to improve understanding of the maritime domain, interpretability is very important. Therefore, random forests have many advantages that are beneficial to this thesis including its higher performance and interpretability.

## E. FEATURES AND PRINCIPAL COMPONENT ANALYSIS

A feature (or attribute) in data analysis is a "property or characteristic of an object that may vary, either from one object to another or from one time to another." [20]. For ships, specifically, some examples of features might include its speed, acceleration, course, or even countable events such as the number of turns it made or times it stopped. The process of finding and extracting useful features for a given classification task benefits from having intimate domain knowledge. These features serve as the input for the classification models described in Section II.D.

However, there may be correlation between several of the features obtained. Since correlation often implies shared information, it may be beneficial to transform the features in order to decorrelate them prior to introducing them to the classification model. One such way to decorrelate features is through principal component analysis (PCA). Before proceeding into discussion of the PCA, it is also noted that sometimes PCA is also referred to as the discrete Karhunen-Loeve transform (DKLT).

The expression of features with orthogonal or uncorrelated basis functions "makes the representation efficient and mathematically convenient" [32]. A basis function is an element within a set that can be used to represent all functions within a function space. A more familiar transformation is the Fourier transform, which uses sinusoids of various frequencies as basis functions. Thus, the PCA transforms a set of features into uncorrelated "principal components." In the context of data analysis, PCA is often used for two main reasons: decorrelating features and dimensionality reduction.

In PCA, the basis functions used to decorrelate the signals are the eigenvectors matrix, $\Phi$, of the correlation matrix, $R_{xx,}$ of the signals or feature set, $x$ [32]. Therefore, the new principal components, $k$, can be derived from the feature set by taking a matrix of the eigenvectors of the correlation matrix and multiplying it with the original feature set. Thus, the principal components are:

$$Principal\ Components = k = \Phi x \tag{12}$$

Therefore, the original feature set $x$ is transformed into a new set of uncorrelated principal components.

It is important to note that these principal components may either boost performance of a classification model or not. Conversely, the correlated information can be important for the ability of a model to distinguish between classes. Therefore, employment of PCA should be tested on a case-by-case basis to determine the effect it will have on the performance of the model in question.

# III. SHIP TYPE AND BEHAVIOR TAXONOMY

This chapter is organized into two sections. The first section lists and explains various ship types as delineated in the AIS data. Additionally, the first section describes several general shipping trends in the Baltic Sea, which is the primary area of study for this thesis. The second section gives a comprehensive analysis of the six most common ship types in the Baltic Sea dataset including their most common behaviors. This section develops a taxonomy of ship behavior to be used for classification in this thesis.

One major goal of this thesis is to be able to develop an algorithm that automates the characterization of ship behavior across multiple types of ships. However, rather than simply beginning with the algorithmic analysis of data, it is necessary to gain an understanding of how each type of ship typically operates and maneuvers in the seas—particularly in the Baltic Sea, which is the use case for this thesis. A better understanding of characteristic ship behaviors will also help inform machine learning methods for detection or classification. Having domain knowledge of ship types and their corresponding operations will thus also inform future work in the selection of machine learning models.

## A. SHIP TYPES DESCRIPTION

There are many ways to classify vessels including by size, capability, or operation type. However, since this thesis aims to use AIS data to develop MDA, a comprehensive listing of ship types is found in the encoding of AIS itself. Additionally, since the machine learning model will be using AIS data in to help mine the data for information, it is convenient to classify ship types using the same types of values that AIS will enumerate. AIS distinguishes between ships that are conducting normal operations (such as cargo ships) and vessels that are engaged in particular activities (such as towing). A full list of ship types used in AIS [33] is shown in Table 3.

Table 3. Ship types in AIS encoding. Source: [33].

| **Normal Ship Type** | **Engaged in Activity Type** |
|---|---|
| Wing-in-Ground | Towing (ahead/alongside) |
| High-Speed Craft | Towing (astern) |
| Passenger | Dredging |
| Cargo | Diving |
| Tankers | Military |
| Pilot | Sailing |
| Search and Rescue | |
| Tugs | |
| Tenders | |
| Anti-Pollution | |
| Law Enforcement | |
| Medical Transport | |
| Pleasure Craft | |

This distinction built into AIS encoding will also help inform machine learning models of ship behaviors since the entire classification built into the "engaged in" column can represent ship behaviors itself. Thus, the descriptors given for vessels in the "engaged in" column can be used to build a supervised dataset potentially for classification or other uses. For instance, if a vessel is currently noted as towing then this information could be used to build a dataset to characterize the towing behavior.

Additionally, it is also important to recognize that not all ships exist in equal proportions or frequencies. This is important to better understand AIS datasets and to get a realistic picture of the types of analysis that can be done. An analysis of the distance that ships travel within the Baltic Sea also gives intuition towards both the number of ships in the region and the frequency at which they operate. A visualization of distance traveled in the Baltic Sea between January 2011 to December 2019 is shown in Figure 12.

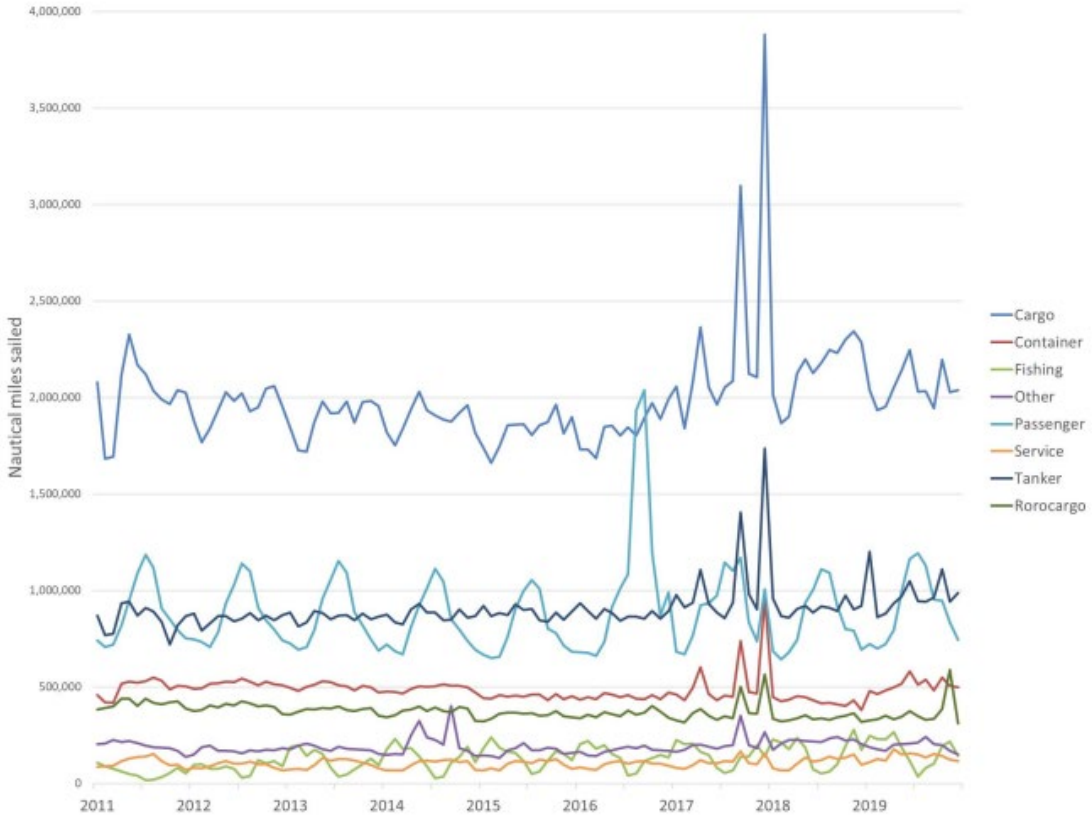



Figure 12. Distance traveled in the Baltic Sea by ship type. Plot determined from HELCOM AIS data. Source: [34].

Thus, from Figure 12, the Baltic Sea shipping traffic is dominated by cargo ships and then followed by tankers and passenger ships. Additionally, it is interesting to note that passenger ships follow a rather sinusoidal cycle of activity, which is likely due to increased demand in the summer months in comparison to the winter months. Thus, future work should keep this ship type frequency in mind for data collection or analysis.

Additionally, while ships may occupy the Baltic Sea at different frequencies or operating tempos, it is also apparent that different types of ships may visit different ports, take alternative routes, or occupy different bodies of water for different periods of time. An analysis of ship routes in the Baltic Sea for the three most common ship types is shown in Figure 13.

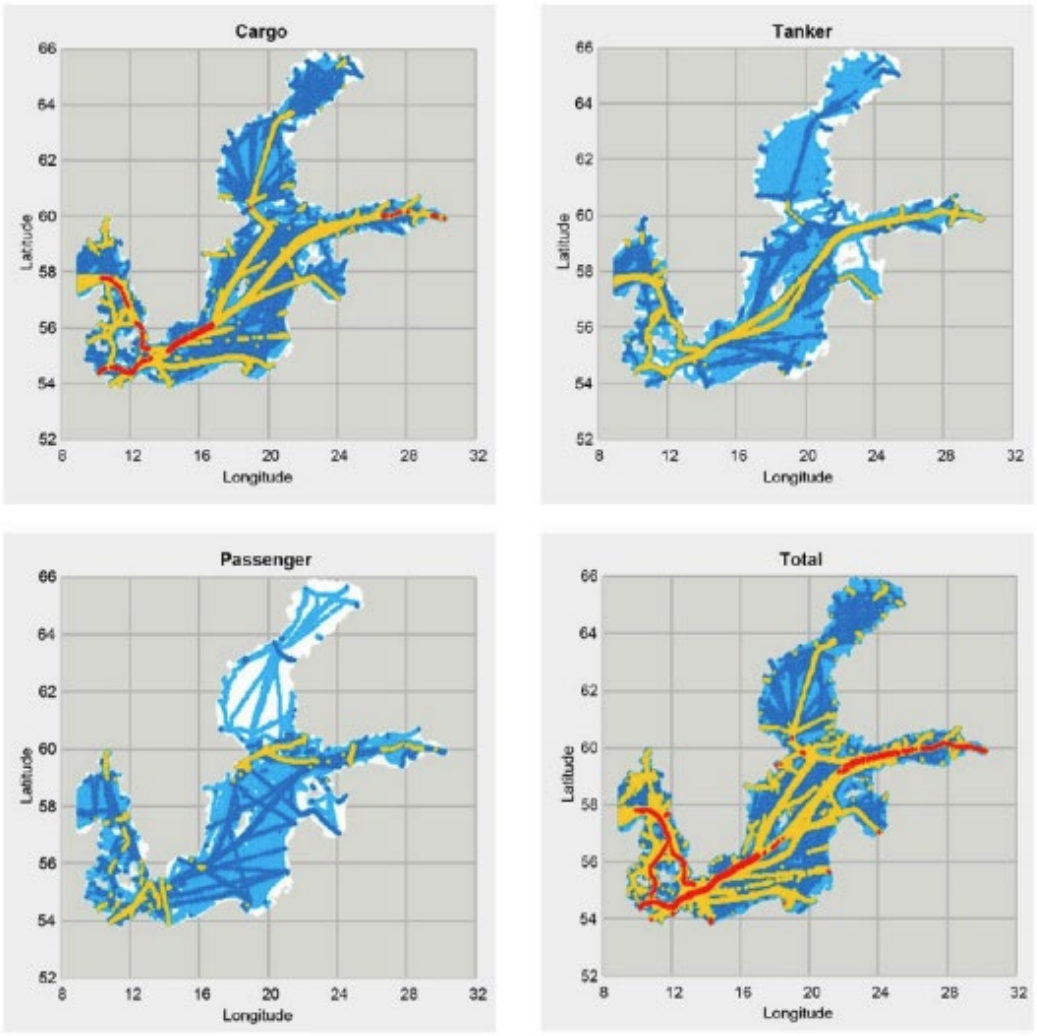



Figure 13. Typical shipping intensity and routes by ship type in the Baltic Sea in 2013. Color coding: white = no vessels, light blue = 1–99 vessels, dark blue = 100–999 vessels, orange = 1000–9999 vessels, red = >10,000 vessels. Source: [35].

As seen in Figure 13, passenger ships tend to take much shorter routes than either tankers or cargo ships. While cargo ships tend to use much of the open water in the Baltic Sea, tankers tend to avoid the northern part of the Baltic Sea. Therefore, each of the three most common ship types tend to take different routes, visit different ports, and occupy different areas of water. Thus, it will be important to understand this when conducting behavioral analysis of ships as different types of ships will not necessarily follow the same

routes as others. In this way, it is important not to make false conclusions on perceived anomalous behavior when what is typical of one type of ship is not typical of another.

## B. SHIP BEHAVIOR TAXONOMY BY SHIP TYPE

To gain a better understanding of ship behavior, it is necessary to differentiate between the many common types of ships. The following detailed description of ship types is certainly not exhaustive nor is it necessarily truly predictive of future trends in maritime activity. Rather, this taxonomy serves as a baseline for future researchers who desire to start studying ship behavior and develop their intuition. Intuition and domain knowledge often inform machine learning models and methodologies.

Before delving into behaviors that are specific to different ship types, it is important to note a behavior common to all ships: anchoring. An example of a ship at anchor is shown in Figure 14.

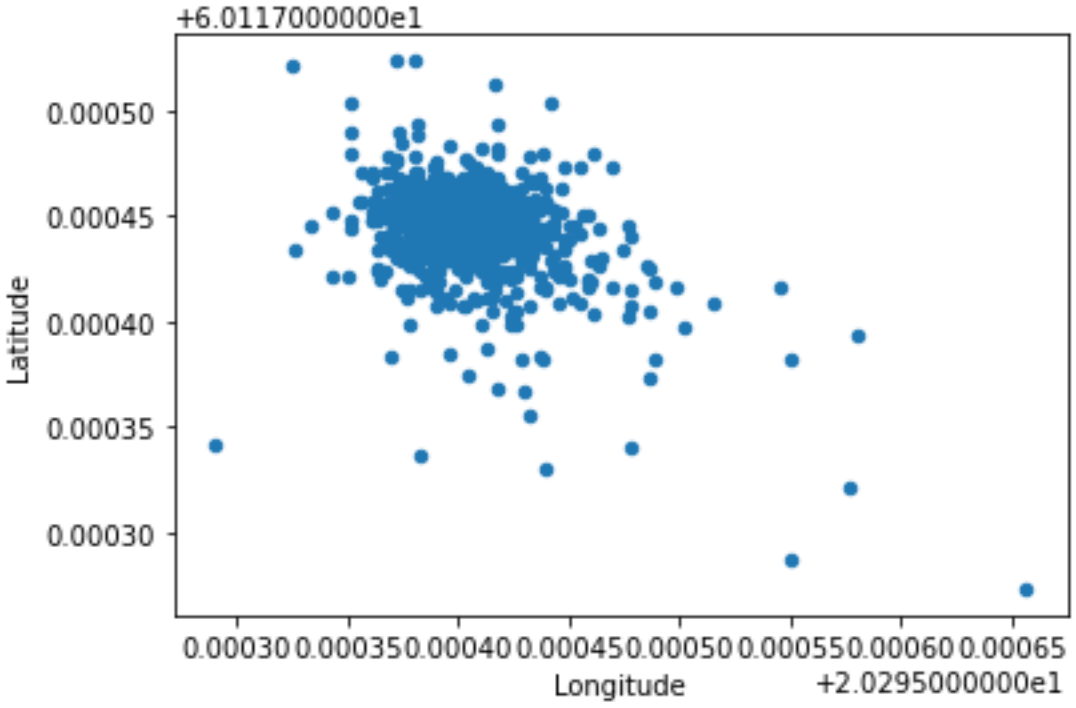



Figure 14. AIS data of a ship at anchor.

A ship at anchor and transmitting AIS data is characterized by near zero speeds, and rather noisy looking position coordinates in a very small region. The noise that appears like a shotgun blast may be caused by several factors including inaccuracies with the location measuring technology or even by slight swaying of the ship either on pier or in open waters anchored. Regardless, this is an activity that is shown by all ships and should be considered in any model attempting to describe ship behavior in general.

### 1. Cargo Ships

Cargo ships transport heavy goods and materials from one port to another. In fact, 90% of all international trade comes through the maritime domain [36]. Thus, the global economy is indeed a maritime economy that relies heavily upon cargo ships. Additionally, the number of ships navigating through the Baltic Sea has decreased despite increasing trade volumes—demonstrating a trend towards larger vessel sizes [37]. Larger vessel sizes are certainly much less maneuverable than smaller ones. They cannot make quick alterations to course nor stop quickly. As a result, ships with such larger mass would be expected to take slow and calculated movements. Additionally, bigger ships require more space for maneuvering, not only in ports (anchorage) but also while on route [37]. Thus, movement through smaller areas should be much more predictable since cargo ships must follow a particular path to avoid grounding or collisions. This has not changed the fact that the highest proportion of accidents the Baltic Sea in 2019 (36.9%) has involved cargo ships [34] as shown in Figure 15.

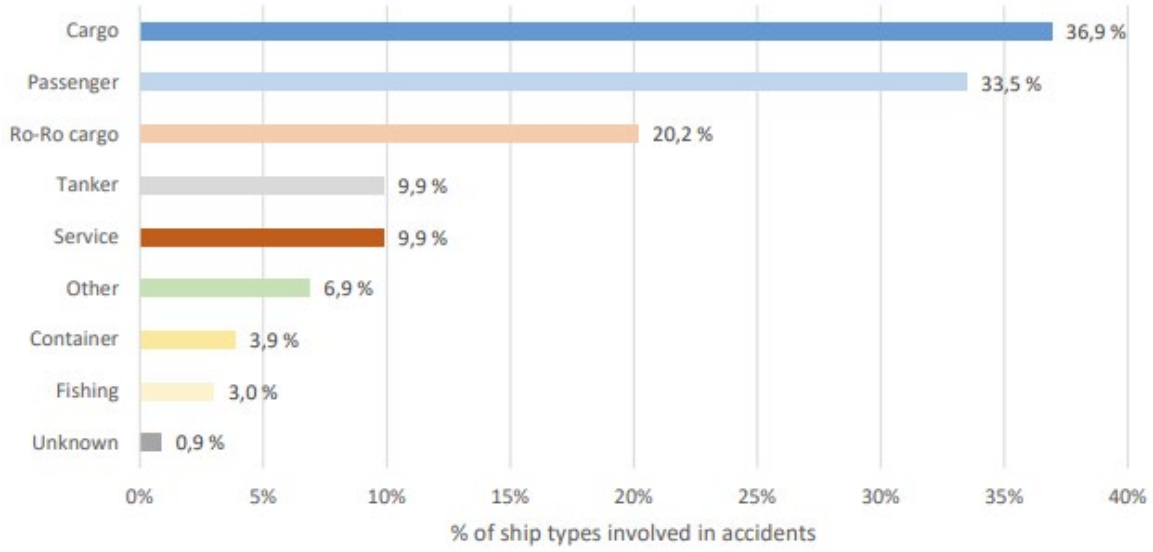



Figure 15. Proportion of ship types involved in accidents in the Baltic Sea in 2019. Source: [34].

Thus, it clear that shipping accidents can have great impact on the global economy and global supply chains. It is therefore important to be able to determine early if a ship is more susceptible to an accident. This, combined with the fact that cargo ships are becoming larger in constrained waters means that any deviation from typical behavior could have severe implications. As a result, future work can focus on anomaly detection of cargo ships for collision avoidance.

One type of activity that is often conducted by cargo ships is long-term transiting. Several other ship types also engage in this activity, but the purpose of cargo ships indeed is for long distance transportation of goods. An example of a long-term transit is shown in Figure 16.

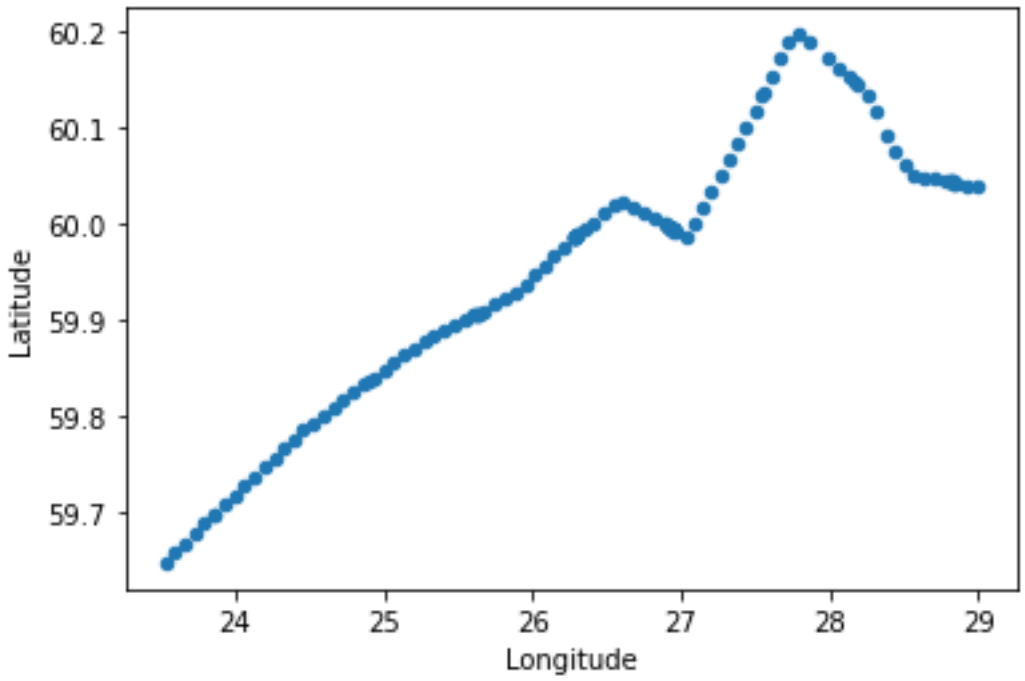



Figure 16. AIS data of a cargo ship during long-term transit.

A long-term transit can be characterized by long straight tracks segmented by deliberate turns. Thus, it would be a relatively good assumption that ships will be traveling at relatively higher speeds during this transit and keep a relatively consistent pacing throughout.

### 2. Tanker Ships

Tankers are similar to cargo ships except that they carry large quantities of liquefied cargo instead of dry cargo. This liquefied cargo can include but is not limited to oil, "alcoholic beverages, hydrogen-based organic compounds, chemicals and even juices" [38]. Tankers do come in a wide variety of sizes and capacities that range from "small self-propelled barges to ultra large crude carriers." Additionally, 30% of all merchant ships are tankers [38]. As a result of the similarity between tankers and cargo ships, the activity of tanker ships is also very similar in that they too also engage in long-term transit, which is seemingly indistinguishable to that of cargo ships.

### 3. Passenger Ships

Passenger ships are merchant ships that specialize in the transportation of passengers or voyagers on either national or international trips. They come in a variety of sizes from yachts all the way up to giant cruise ships [39]. Passenger ships come in two forms—ferries and cruise ships. Ferries are used to transit voyagers on shorter trips. They operate on a regular schedule and have fixed fares [39]. Additionally, they typically travel on the same routes repeatedly with many intermittent stops similar to other forms of public transport such as bus or rail. Cruise ships are larger vessels equipped with luxuries and a variety of amenities. Typically, they encounter longer voyages and travel to "destination vacations." As a result, the ports that cruise ships visit will likely be well known, predictable, and repeatedly visited in the data available. An example of the repetitive activity of passenger ships is shown in Figure 17.

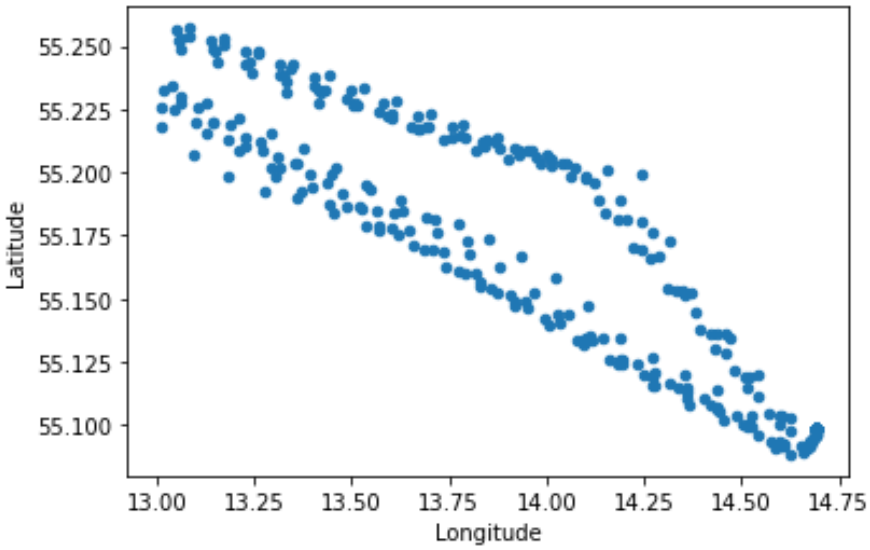



Figure 17. AIS data of a passenger ship during ferrying activities.

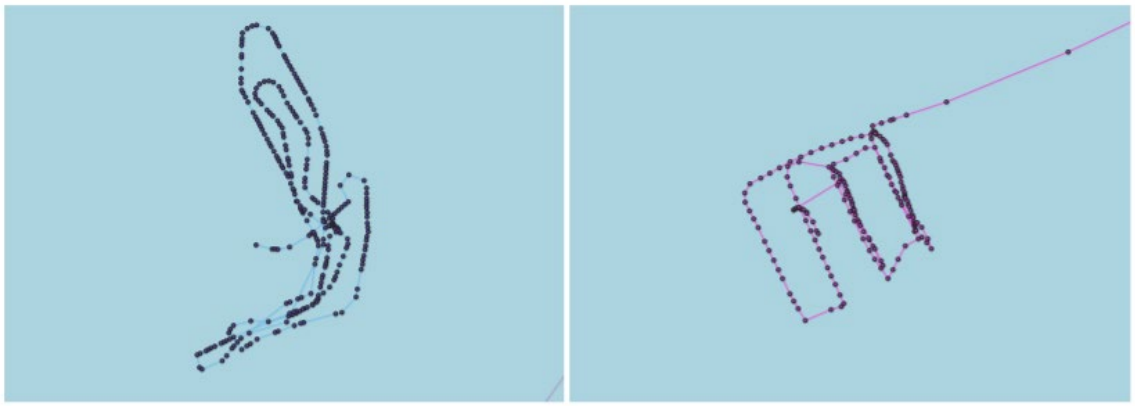
As demonstrated in Figure 17, the AIS data of a passenger ship engaged in ferrying is repetitive in that it is traveling over the same routes again and again, which generates the noisy looking tracks. Additionally, it is important to note that passenger ships are not travelling nearly as far as cargo ships or tankers. However, a key distinguishing

characteristic of ferrying is the fact that its track is repetitively travelled and generates noisy looking cumulative tracks. A cumulative track is what is seen when the AIS data is displayed, and the vessel has made several or many trips through an area already.

### 4. Fishing Ships

Fishing in the Baltic Sea is regulated to the Common Fisheries Policy [40]. This type of regulation is not uncommon to many areas of the world. Thus, it is a matter of legal interest that fishing vessels are engaged in lawful fishing activity in areas where they are allowed to be fishing. Detection of unlawful fishing activity is therefore another application of machine learning models that seek to enhance MDA.

Fishing can be separated into two separate yet similar methods—trawling and longlining. Fishing vessels engaged in trawling "use a fishing net located in the stern of the boat, called trawl, which is dragged through the water" [13]. Throughout trawling, vessels typically sail with lower steady speeds than they typically would. Fishing vessels engaged in longlining "set multiple fishing lines with baited hooks attached to them called snoods" [13]. When vessels are setting their lines, they typically travel at a constant speed but once they are set the vessel will then travel slowly to drift with the lines. An example of longlining and trawling is shown in Figure 18.



(a) The movement pattern of a trawling trajectory. (b) The movement pattern of a longlining trajectory.

Figure 18. AIS data of fishing vessels engaged in trawling and longlining. Source: [13].

However, there are a great number of similarities between fishing vessels engaged in trawling and longlining. In both activities, fishing vessels will travel at slower steady speeds while making many course changes. Additionally, both activities may take place over "several hours or even days" [13]. As a result of the similarity of these activities, it is possible to combine both subclasses into the larger overall class of fishing. An example of fishing in AIS data is shown in Figure 19.

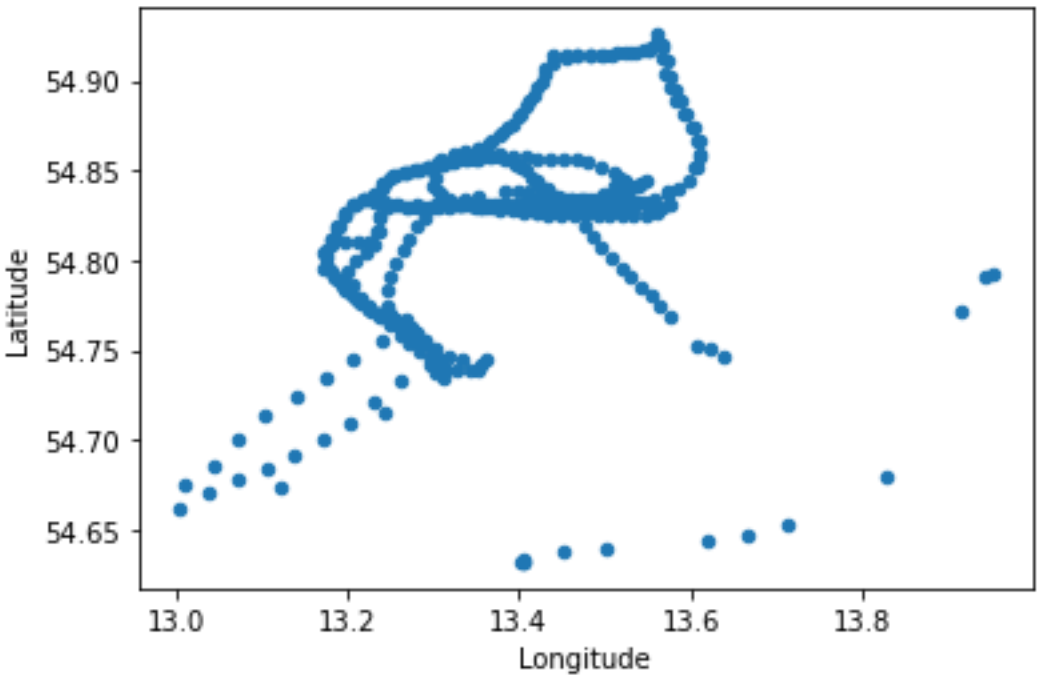



Figure 19. AIS data of fishing vessels engaged in fishing.

Thus, for this thesis, fishing is be generalized to include both trawling and longlining activities since both involve very similar behaviors. However, an area of future work could include the differentiation between these two activities in the presence of many other activities, as extension of the work presented in [13].

### 5. Tugboats

A tug or tugboat is a secondary support ship that helps in the "mooring or berthing operation of a ship by either towing or pushing a vessel towards the port" [41]. However,

their hull design typically makes it dangerous for them to sail in open ocean waters [41]. Rather, they operate in more constrained waters where they can supply their support. Large vessels cannot be maneuvered by their own and may need tugboats to help them navigate through narrow water channels. Additionally, tugboats can also serve as salvage boats, icebreakers, or firefighting ships. Thus, tugboats can serve in a variety of support roles. Due to their nature, tugboats often demonstrate cooperative behavior with larger vessels such as cargo ships or tankers. The ability to detect cooperative behavior must first involve the knowledge that two vessels are in close proximity in both space and time. Once two ships are detected to be close enough together for cooperation to be possible, additional features need to be extracted from the AIS data of each ship in order to determine cooperation. As a result, future work on the cooperative nature of such behavior could help develop greater understanding of the maritime domain.

### 6. Pilot Ships

A pilot ship is a vessel that is used to ferry helmsmen or pilots from harbors to larger ships anchored in the harbor that need piloting [42]. Pilot vessels are designed to be durable and strong through many types of inclement weather and can reach speeds up to 32 knots [42]. Thus, pilot vessel activity is characterized by routine short trips from a common location within the harbor to nearby locations where other ships are anchored. An example of a pilot ship trajectory is shown in Figure 20.

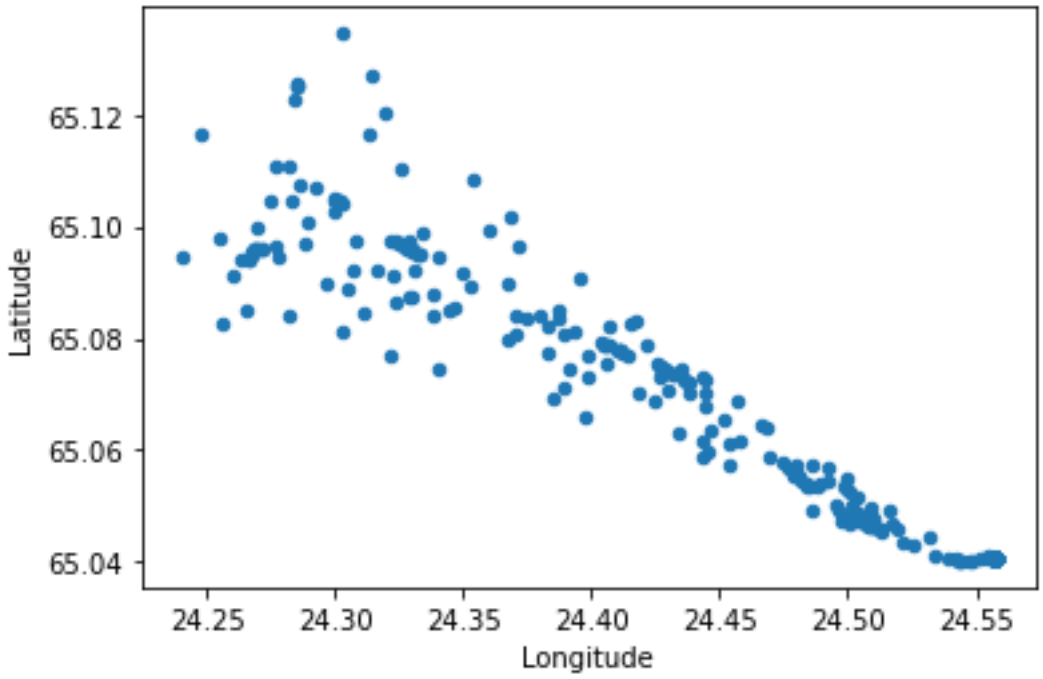



Figure 20. AIS data of a pilot ship during ferrying activities.

Thus, as expected, pilot vessel activity appears to originate from a common origin while making many similar trips of a short distance out into the harbor. Therefore, the activity between ferrying and piloting appear very similar except that piloting typically includes one singular origin point to many different destinations while ferrying involves common origin and destination points. Therefore, any attempt to differentiate between the two activities should utilize a feature that attempts to capture this fact.

THIS PAGE INTENTIONALLY LEFT BLANK

# IV. DATA PREPARATION

This chapter is organized into five different sections. The first section describes the geographical filtering that was applied to the dataset to provide focus to this study. The second section describes an autonomous method that utilizes the DBSCAN clustering algorithm to filter out anchorages from the dataset. The third section describes a method for sectoring and downsampling the dataset to achieve desirable algorithmic runtime characteristics. The fourth section describes mathematically the way in which twelve different features were extracted from the spatiotemporal data contained in the AIS data. Finally, the fifth section describes principal component analysis as a method for preprocessing the feature set for potentially better classification performance.

When it comes to any type of data analytic workflow, often, much of the effort is in the steps leading up to the actual modeling and analysis itself. Datasets are often messy and have many errors that must be corrected before meaningful analysis can be conducted. Some common issues can include but are not limited to human error, limitations of measuring devices, flaws in data collection, missing values, missing data objects, spurious objects, and duplicate objects [20]. As a result, it is often necessary to clean, wrangle and visualize data before performing any machine learning-based analysis or model construction.

## A. DATA DESCRIPTION AND GEOGRAPHICAL FILTERING

In order to develop an automated approach for the analysis of ship behavior from AIS data, an area of the world needed to be chosen in order to give focus to the study. However, regardless of the area chosen, the analysis that follows can be conducted in any region or body of water. For this thesis, the Baltic Sea was chosen as the area of interest. The region and its major ports are shown in Figure 21.


Figure 21. Political map of the Baltic Sea highlighting major ports and cities. Source: [43].

The Baltic Sea covers an area of 377,000 km$^2$ and borders nine countries—Denmark, Estonia, Finland, Germany, Latvia, Lithuania, Poland, Russia, and Sweden [43]. Additionally, the Baltic Sea is home to more than 200 ports that see over 30% of European Union exports and imports sail through its waters. Thus, the Baltic Sea is an area of great economic importance. Additionally, the Baltic Sea was chosen because there is one main inlet and outlet that connects solely to the North Sea. As a result, the paths that ships take in the region will clearly be more controlled and organized than if the same analysis was conducted in open ocean. With the variety of economies that occur within the Baltic Sea it is probable that a multitude of diverse ship behaviors can be observed for future characterization.

Thus, the original dataset was constrained to positional data located within the Baltic Sea. This was easily implemented through a geographical filter that filtered out data objects with longitudes less than 13° E or longitudes greater than 29° E. The geographical filter also filtered out data with latitudes less than 53° N or latitudes greater than 66° N. The resulting filtered dataset was then able to be visualized as shown in Figure 22.

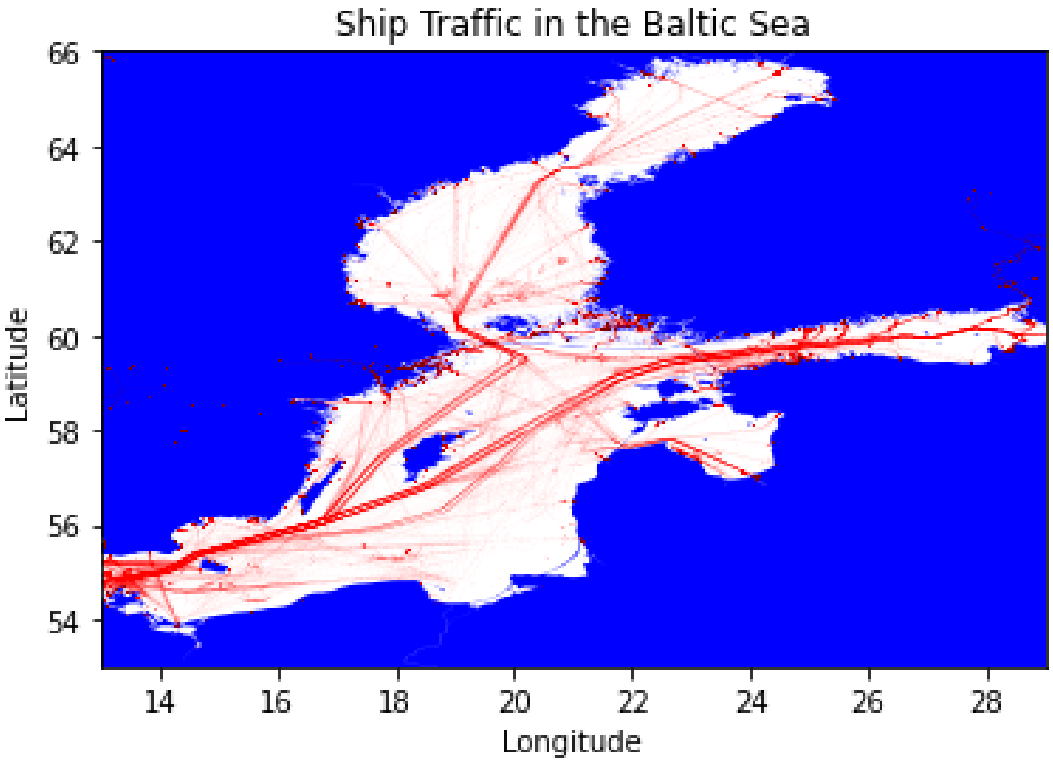



Figure 22. Filtered dataset for ship data within the Baltic Sea.

After the dataset was filtered, it was characterized for further analysis. Within the dataset, there are 2586 unique ships indicated by their unique identifier or MMSI. Additionally, there are 13 048 552 data objects within the dataset that were logged by AIS transponders from November 26, 2013, to January 7, 2014. The kind and number of ship types that were logged in the dataset were characterized in Table 4.

Table 4. Summary of ship types within dataset.

| Ship Type | Frequency |
|---|---|
| Cargo | 1017 |
| Tanker | 320 |
| Fishing | 240 |
| Passenger | 238 |
| Tug | 143 |
| Pilot | 119 |
| Search and Rescue | 73 |
| Law Enforcement | 53 |
| Dredging | 40 |
| Military | 21 |
| Towing | 19 |
| High Speed Craft | 14 |
| Sailing | 9 |
| Tender | 4 |
| Diving | 4 |
| Other/Not Specified | 272 |
| Total | 2586 |

## B. ANCHORAGE AREA FILTERING

An early observation in algorithm development was that if many points were concentrated in one spot, this can create "imaginary" tracks since the similarity of the points causes many lines to appear to originate from the same area. Since each of the anchorage points are very close in proximity, they will generate nearly identical $r$-$\theta$ parameters with the rest of the points. Therefore, since the algorithm developed in Chapter II relies on the density of $r$-$\theta$ parameters, the presence of the anchorages could cause errors in the track identification process. The areas of high concentration, which are typically anchorages for ships, are indicated by the bright red "dots" in the original data as shown in Figure 23.

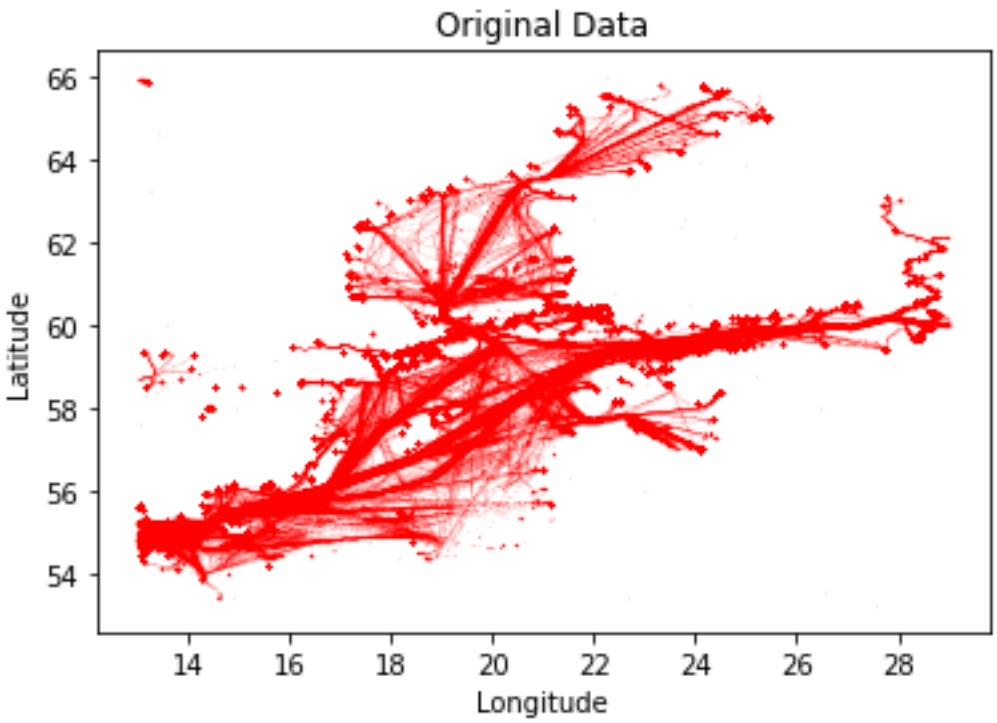



Figure 23. Original ship data with anchorages shown as bright red points.

One way to filter out these points is to utilize the DBSCAN algorithm, which classifies data points located within high density regions as belonging to clusters and classifies the rest of the data points as noise. In contrast to the case described in Section II.C.3, it is desired to detect high-density regions (anchorages) so that they can be discarded in favor of lower-density regions (tracks). As a result, *Eps* was chosen to be incredibly small at *Eps = 0.001* since anchorage points should indeed be very close to one another as ships tend to stay in one area when anchored or in port. Additionally, *MinPts* was chosen to be *MinPts = 10* as there should be relatively large number of points packed together for there to be an anchorage area. The result of filtering for the removal of anchorage areas is shown in Figure 24.

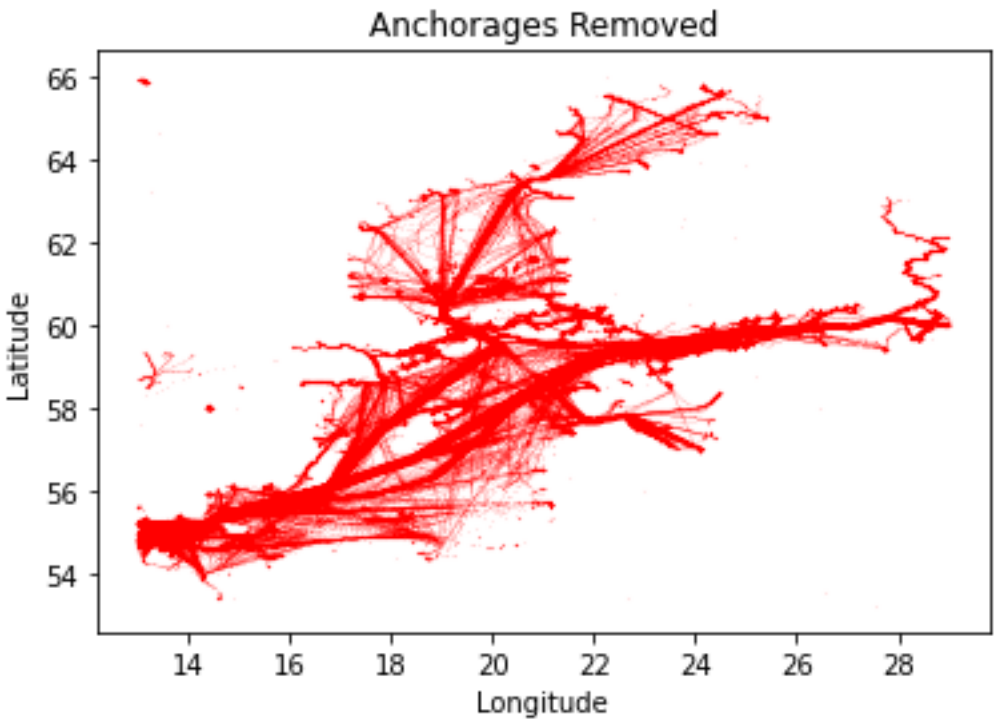



Figure 24. Ship data after filtering for the removal of anchorage areas.

Thus, it appears that the filtering was successful in removing the anchorage areas from the data while keeping the important data that contains the track data, thus enabling further processing to automatically detect the most populous tracks within the Baltic Sea data.

## C. GEOGRAPHIC SECTORING AND DOWNSAMPLING

As discussed previously, this dataset contains over 13 million data objects and covers a vast region of the Baltic Sea. Contained within the Baltic Sea data—as confirmed visually—are many different tracks. Thus, if analysis were to be conducted on the entire dataset there would be a variety of complications.

The first issue rises in the identification of tracks using the entire dataset. Due to the sheer multitude of points, the generated $r$-$\theta$ values will be densely packed together and the identification of unique values will be very challenging. There are many tracks within

the Baltic Sea, however these tracks may be easily discarded for a best fit line through all the data points instead.

The second complication—and perhaps the most challenging—is that for a dataset of 13 million data objects, there would need to be over $10^{13}$ computations for $r$ and $\theta$ values per Equation (4). This does not even account for the run speed of the DBSCAN algorithm. Thus, performing analysis on the entire dataset is too computationally taxing—the results of which may never be received nor used by the operator or decision maker who clearly would desire a timelier result.

### 1. Geographic Sectoring

A solution to both issues is employment of geographic sectoring and downsampling. Geographic sectoring solves the first issue by reducing the area of analysis into "bite-size" chunks, which reduces the number of contained tracks. This simplifies analysis by making the $r$-$\theta$ space sparser and thus the identification of unique, meaningful clusters easier. However, there may be many ways to geographically sector the data. One meaningful way is to break the data up into $N$ rectangular sectors for analysis. These $N$ rectangular sectors would be constructed by $N$ geographic filters similar to the filter discussed in Section IV.A. Once the data has been broken up into geographic sections, analysis can be conducted within each region independently. Then once analysis on all the regions is complete, the results can be joined together to form a complete analysis of the region of interest, the Baltic Sea region in this thesis.

### 2. Downsampling

Geographic sectoring allows for meaningful results to be found on smaller regions. However, it is possible, and likely, that the number of data objects contained in sectored regions will still be prohibitively large, making further analysis too computationally intensive, and thus producing an algorithm that is too slow to provide utility to an operator or decision maker. A solution to this problem, which improves run time on a sectored region, is downsampling the sectored data to a reduced number of data points. Downsampling to a set number guarantees that the algorithm makes the same number of computations each time and thus guarantees a run time. Additionally, downsampling

should not impact the performance of the algorithm as the denser areas (corresponding to tracks) should still be denser than noisy points after random downsampling. This technique is particularly useful for an operator or decision maker who needs to make a real time decision as it guarantees an analytical result within a usable period of time. Then, after geographic sectoring the data is standardized as explained previously in Section II.C. After downsampling, the data can then be analyzed for tracks. Once the analysis of the sectored areas is complete, the results can be joined together for a cumulative result. An overview of the algorithm for the automation of the identification of tracks is shown in Figure 25.

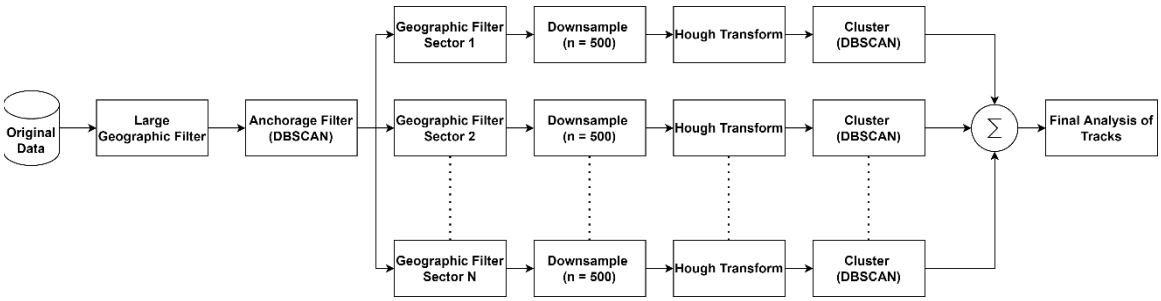



Figure 25. Track identification process with data cleaning.

Thus, through the process depicted in Figure 25, the total body of data can be analyzed for tracks in a computationally efficient way. The choice of the number of geographic subsectors is best chosen for a given situation and region. In this way, domain knowledge along with some basic data visualization can help inform later machine learning decisions.

## D. FEATURE EXTRACTION FROM SPATIOTEMPORAL DATA

Spatiotemporal data is data that includes information about an object location in both time and space. For vessels, this typically includes its latitude/longitude coordinates as well as a UTC timestamp. One advantage of spatiotemporal data is that it can be derived from a variety of sources beyond AIS. Such information can be gained from satellite imagery [44], shipborne radar [45], among many other methods that do not rely upon a cooperative actor. Thus, the ability to derive features and classification results from spatiotemporal data alone is a highly effective technique especially when looking for

uncooperative actors. These uncooperative actors could be ships conducting illicit activities and thus hiding their true intentions in the AIS data. Thus, deriving features only from the spatiotemporal data from AIS or other intelligence sources may help shed light on their true intentions.

In the navigational domain, it will be useful to define the distance ($D_k$) between two sets of longitude/latitude coordinates. One common measure of the distance between such positional coordinates is the Haversine distance. The Haversine distance formula gives the great-circle distance (shortest path distance on a sphere) from their longitude and latitude coordinates [46]. The Haversine distance formula between two points $P_{k+1}$ and $P_k$ respectively with latitudes $\varphi_{k+1}, \varphi_k$ and longitudes $\theta_{k+1}, \theta_k$ is shown in Equation (13).

$$D_k = 2r\sin^{-1}\left(\sqrt{\sin^2\left(\frac{\varphi_{k+1}-\varphi_k}{2}\right)+\cos(\varphi_k)\cos(\varphi_{k+1})\sin^2\left(\frac{\theta_{k+1}-\theta_k}{2}\right)}\right) \quad (13)$$

$D_k$ is the great-circle distance between points $P_{k+1}$ and $P_k$. The $r$ is the radius of the Earth, given as 3443.92 NM. Thus, knowledge of two position reports can also inform the distance between those two points, which is a fundamental quantity used to derive other features.

Another fundamental quantity that can be derived from two positional reports is the bearing ($B_k$) between those points [47]. The bearing between two positions can be calculated as:

$$B_k = \arctan 2(\cos(\varphi_{k+1})\sin(\theta_{k+1}-\theta_k), \cos(\varphi_k)\sin(\varphi_{k+1}) - \sin(\varphi_k)\cos(\varphi_{k+1})\cos(\theta_{k+1}-\theta_k)) \quad (14)$$

Another fundamental quality that can be computed is the average speed that a ship took between two positional reports. There exists *n-1* speed reports for a ship with *n* positional reports. The average speed $s_k$ of a vessel that was observed at position $P_k$ at time $t_k$ and position $P_{k+1}$ at time $t_{k+1}$ is shown in Equation (15).

$$s_k = \frac{\text{Haversine}(p_{k+1}, p_k)}{t_{k+1}-t_k}, k = 1, 2, ..., n-1 \quad (15)$$

The distance and bearing between two points are key measures that can be used to define a great multitude of potentially useful features. From here forward, the Haversine distance is abbreviated as $D_k$ and bearing is abbreviated as $B_k$. A list of derived features for a ship trajectory of length $n$ positional reports is shown in Table 5.

Table 5. Summary of features extracted from spatiotemporal ship track data.

| **Feature** | **Description** | **Equation** |
|---|---|---|
| Mean Speed | Average speed of a ship through its course | $\bar{s} = \frac{1}{n-1}\sum s_k$ |
| Std Speed | Standard deviation of speed of a ship through its course | $\sigma_s = \sqrt{\frac{\sum (s_k - \bar{s})^2}{n-1}}$ |
| Skew Speed | Skewness of speed of a ship through its course | $skew = \frac{\sum (s_k - \bar{s})^3}{\sigma_s^3 (n-1)}$ |
| Kurt Speed | Kurtosis of speed of a ship through its course | $kurt = \frac{\sum (s_k - \bar{s})^4}{\sigma_s^4 (n-1)}$ |
| Displacement | Distance between first and last position | $D = \text{Haversine}(p_n, p_1)$ |
| Number of Turns | A count of all turns greater than 60°. | $NumTurns = \sum (\lvert B_{k+1} - B_k \rvert > 60)$ |
| Normalized Number of Turns | Number of Turns normalized for the length of the track | $NormNumTurns = \frac{1}{n-2} NumTurns$ |
| Number of Speed Changes | Number of times there is a speed change greater than 4 kts from one time to another | $NSC = \sum C_k, C_k = \begin{cases} 1 \text{ if } \lvert s_{k+1} - s_k \rvert > 4 \\ 0 \quad \text{otherwise} \end{cases}$ |
| Normalized Number of Speed Changes | Number of times there is a speed change greater than 4 kts from one time to another normalized for the length of the track | $NormNSC = \frac{1}{n-1} NSC$ |

| Feature | Description | Equation |
|---|---|---|
| Number of Stops | Times the Ship stops and starts back up during its course | $NS = \sum C_k, C_k = \begin{Bmatrix} 1, \text{stop if } \lvert s_k \rvert < 0.1 \text{ and now moving} \\ 0, \text{moving if } \lvert s_k \rvert > 1 \text{ and now stop} \end{Bmatrix}$ |
| Accumulated Angle | Sum of all bearing changes if ship is moving | $AA = \sum \lvert B_{k+1} - B_k \rvert \text{ if } s_k > 1$ |
| Accumulated Distance | Sum of all distances throughout trip | $AD = \sum D_k$ |

It is important to note that other more creative, and possibly more expressive features can be extracted from the spatiotemporal data. The usefulness of a feature for classification tasks is left best to the discretion of one who has domain knowledge. After all, features that may be important to identify anchoring may not be the same features that may be important to identify fishing. In fact, this thesis shows that this hypothesis can be confirmed, especially through an interpretative model such as a random forest model, which allows for the relative importance of features to be visualized for a given classification task.

The expressivity of a feature can also be evaluated by its correlation to other features. If two features are highly (positively or negatively) correlated, then they share a linear association with one another. Thus, it is highly desirable for features to be as uncorrelated as possible since uncorrelation implies new information learned about an object. A visualization of the correlation between the twelve features described in Table 5 is shown in Figure 26.

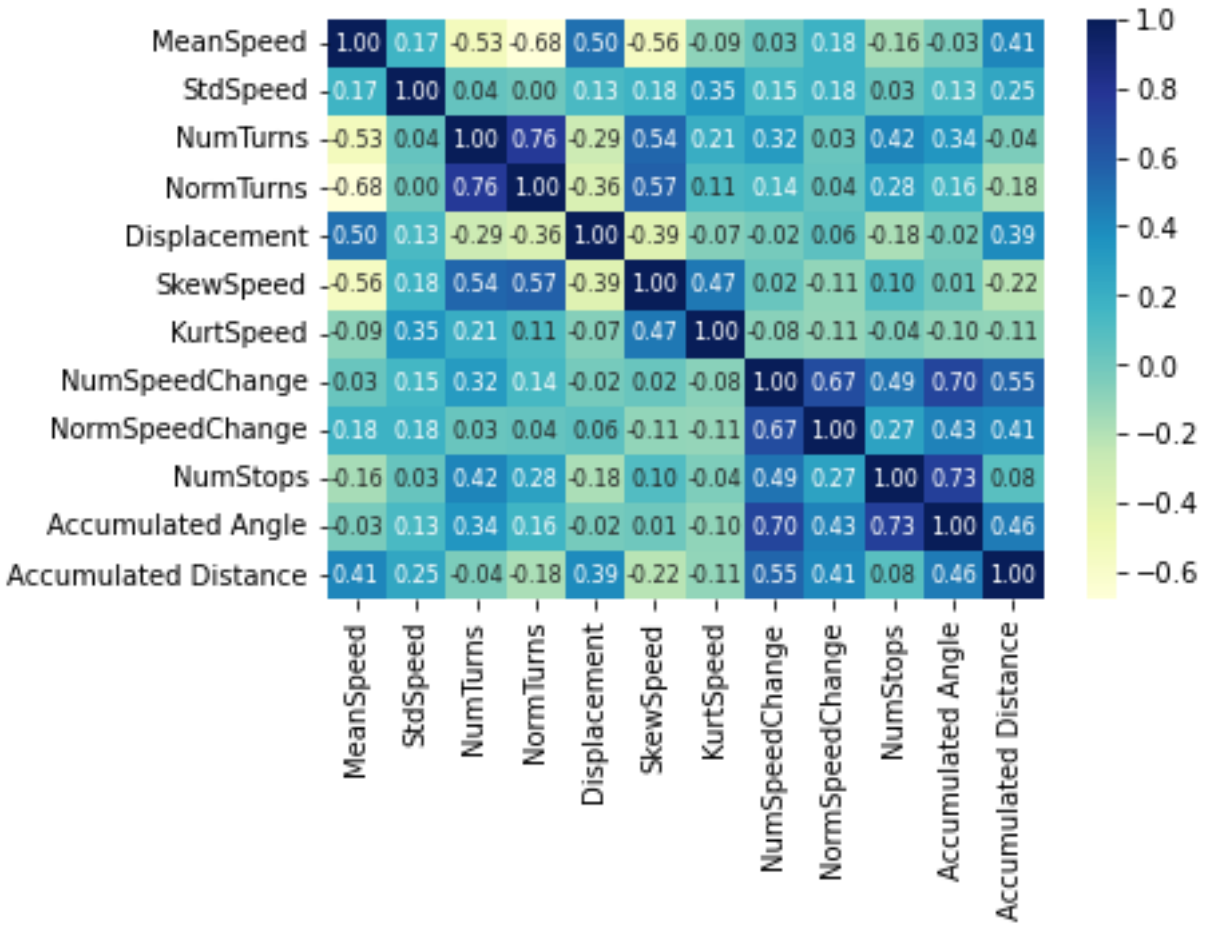



Figure 26. Correlation heatmap between features extracted from spatiotemporal data.

Thus, many of the features are uncorrelated with one another and contain "new" information about the ship traveling in the spatiotemporal domain. However, even features that are highly correlated such as "accumulated angle" and "number of speed changes" with correlation = 0.84, may still have new information that may be useful for a machine learning model to make predictions. As a result, analysis and refinement of the model performance itself is necessary before removing features from the ensemble. The creation of features is a highly iterative process that receives direct feedback from model performance.

## E. PRINCIPAL COMPONENT ANALYSIS APPLICATION

As shown in the correlation heatmap of Figure 26, there exists correlation between some of the features. Correlation often indicates shared information and as a result it may be potentially fruitful to transform features to decorrelate them before applying a machine learning model. As discussed in Section II.E, one such way to decorrelate features is through PCA.

PCA was then applied to this feature set to determine its effectiveness in boosting the performance of a classifier attempting to distinguish between different ship behaviors. The resulting correlation matrix is shown in Figure 27.

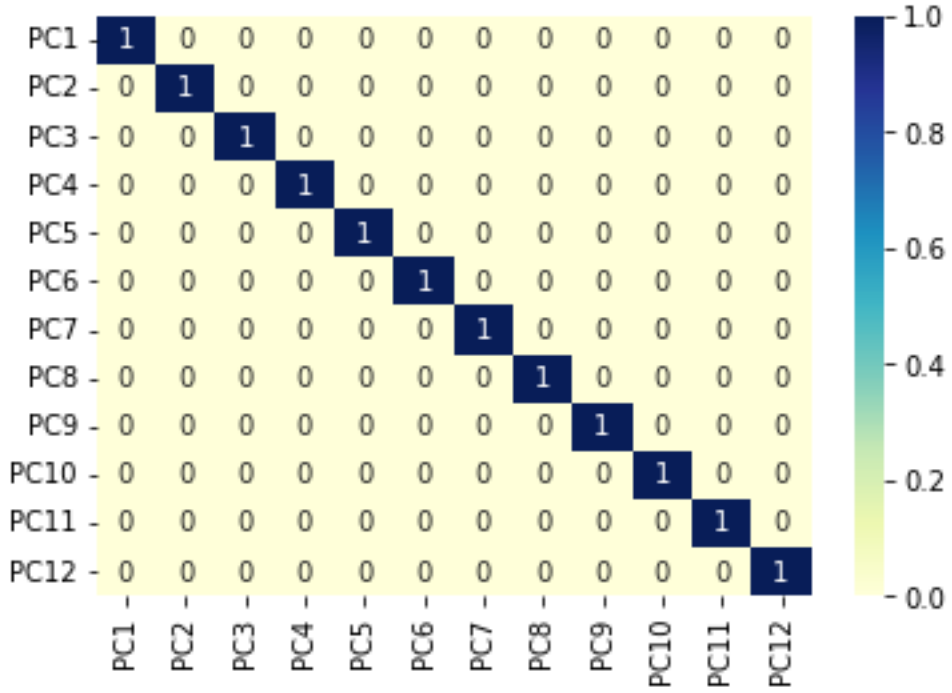



Figure 27. Correlation heatmap principal components derived from features using PCA.

As expected, the correlation matrix of the principal components is diagonal, indicating that the principal components are indeed uncorrelated with one another. Therefore, the implementation of PCA was successful.

Additionally, if sorted in decreasing order, the magnitude of the eigenvalues also demonstrate the relative importance of the transformed components. This is because the eigenvalues act as a multiplier to the eigenvectors. Therefore, eigenvectors with a larger corresponding eigenvalue are emphasized over eigenvectors with a smaller eigenvalue. Thus, the total energy in each principal component can be analyzed for importance. If the goal of PCA is to accomplish feature reduction (also known as dimensionality reduction), the principal components with the smallest energy (smallest eigenvalues) can be removed. However, this may or may not help a specific machine learning model. If a machine learning classifier is relying on the details (small differences) to decide between two classes, then the removal of the small energy components may harm the model. Alternatively, removing the small details could help the model by reducing the overfitting effect. In this case, it is situationally dependent and thus both methods should be tried in order to determine the best possible outcome. An example of explained variance for this dataset is shown in Figure 28.

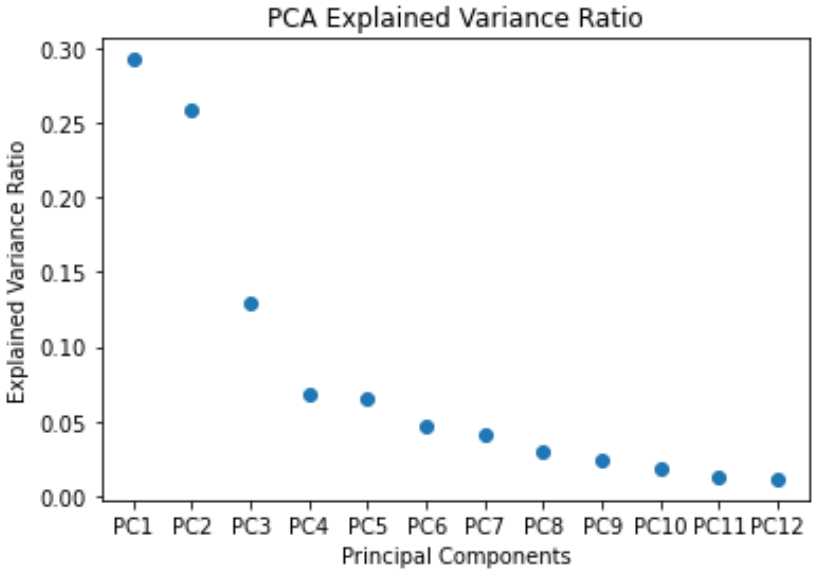



Figure 28. Explained variance ratio for the principal components derived from the extracted spatiotemporal features.

From Figure 28, it appears that most of the explained variance or energy is contained within the first five principal components. After the fifth component, there is a notable drop off in variance. However, as discussed earlier, though this value is small, it may still influence the decision performance of a model and thus should be considered during model selection.

Before continuing to data analysis and results, it is first necessary to understand the physical meaning of principal components. Features (in this case mean speed, standard deviation of speed, etc.) clearly have a meaning. Principal components that are derived features *typically* do not. This is because principal components, after all, are just linear combinations of the original features derived in such a way so that they are uncorrelated. Therefore, it is important to understand that while the model performance may improve from use of PCA, interpretability of that model may become more unclear.

THIS PAGE INTENTIONALLY LEFT BLANK

# V. DATA ANALYSIS AND RESULTS

In this chapter, the results of both the track identification and ship behavior classification are presented. Both processes resulted in visualizable and interpretable results. In an area such as MDA, it is necessary that the results from a machine learning model be both predictive and interpretable. Since the goal of this research is to advance understanding of the maritime domain, the interpretability of the models is paramount.

## A. TRACK IDENTIFICATION

The track identification process shown in Figure 25 was utilized to analyze the data within the Baltic Sea. The data was sub-sectored into 100 geographic regions by equally dividing the width and height of the Baltic Sea region by 10 each so that the longitude resolution is 1.6° and the latitude resolution is 1.3°. Thus, the entire Baltic Sea region was subdivided into geospatial blocks with a width of 1.6° and height of 1.3°. Next, the resulting regions were downsampled so that they would have a size of 500 data objects in order to guarantee run time for the algorithm.

Next, each sub-sectored block of data had its $r$-$\theta$ values computed through the Hough transform. Lastly, the resulting $r$-$\theta$ space was clustered using the DBSCAN algorithm in order to identify the most prominent tracks. The individually sub-sectored results were then synthesized together to form the result of the automated algorithm shown in Figure 29.

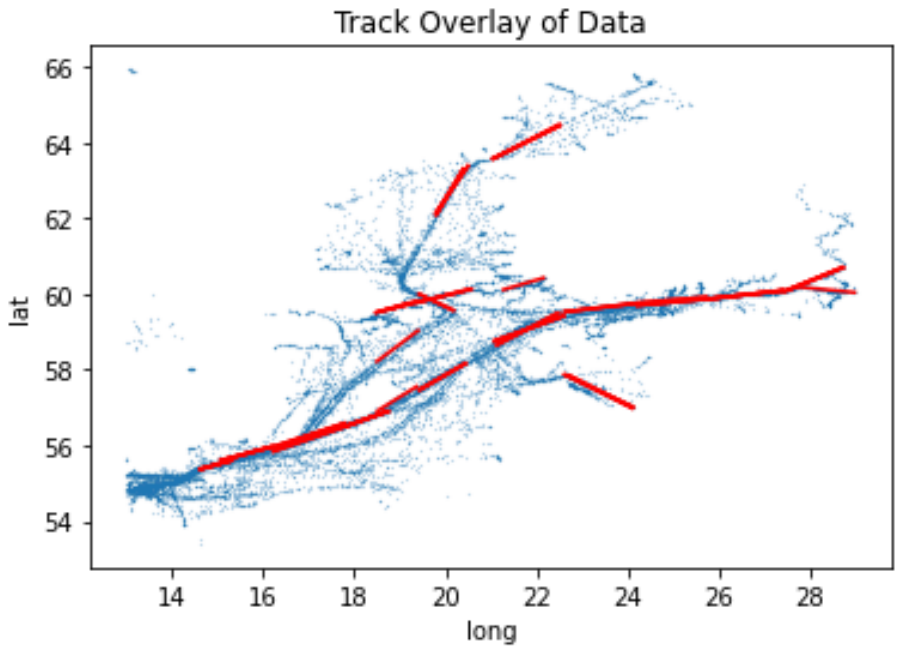



Figure 29. Identification of the most populous tracks in the Baltic Sea using the Hough transform line identification technique.

Thus, the most prominent tracks were identified using the Hough transform on the geographically sectored data. However, there are several design decisions that need to be made in applying this track identification method in the future. The most important characteristic that needs to be decided is the size of the geographic subsections. Choosing subsections too large may cause tracks to be too generalized and lose out on meaningful information. However, choosing subsections too small may give undue importance to tracks that are unimportant and small. This decision is highly dependent on the region of the world being studied and the context for which the investigation is being conducted. Thus, thoughtful analysis of these questions can help inform an operator to select optimal parameters for future applications.

An application of this track identification is to identify new, emerging or changing tracks. The identification of new or changing routes is an important task since it could also identify issues with existing routes or indicate changes in global shipping trends. Regardless, each of these situations carry significant economic or political significance that may be necessary to be known by decision makers.

## B. SHIP BEHAVIOR CLASSIFICATION

As discussed previously, the random forest model was chosen as the best model for behavioral classification in this thesis. In particular, the random forest model offers great interpretability for MDA analysis while maintaining excellent performance. The bulk of this thesis lies in the identification and classification of ship behavior. As a result, this task has been broken up into several parts. First, several binary classifiers were built. Binary classifiers are helpful in that they can be used to identify the most important features for the identification of a particular behavior. Indeed, one of the advantages of the random forest model is that the most important features can be identified. In this way, additional information about the nature of each activity was able to be learned through the binary classifiers before implementing a final multi-class classifier. All classifiers were built using hand-labeled data from the previously described dataset.

Any results of these classifiers, particularly in regards to accuracy, must be evaluated while considering the balance of the dataset as noted in Section II.D.1. Next, features from each ship were extracted from each ship spatiotemporal data contained within the AIS data. These features were then fed into the random forest models that then were able to show the most important features for the model classification scheme. This is true for all the classification models built within this thesis.

### 1. Transiting and Anchoring Binary Classifier

The first binary classifier built utilizes data from cargo ships in order to classify ships that are either transiting or anchored. This dataset has 787 data objects of which about 89.7% are vessels transiting and 10.3% are vessels anchored. The features used for this classification task are: mean speed, standard deviation of speed, number of turns, normed number of turns, skew of speed, number of speed changes, normed number of speed changes, number of stops, and accumulated angle. An example of feature importance is shown in Figure 30.

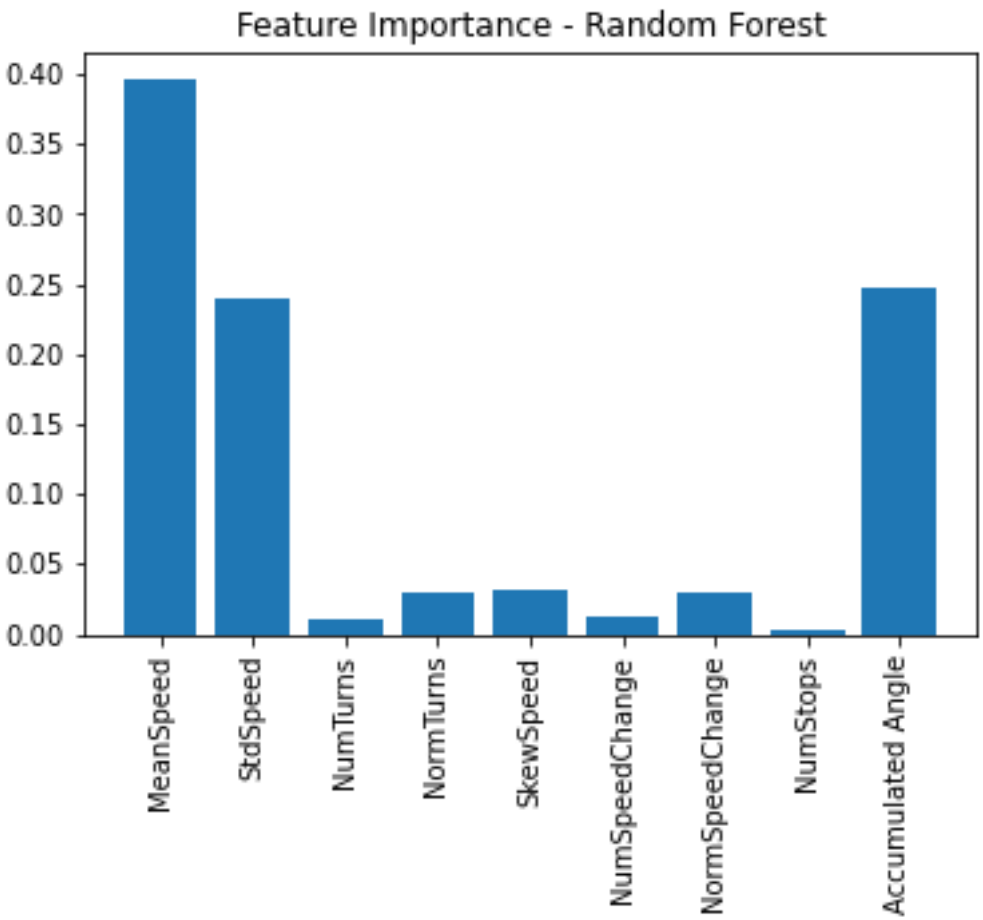



Figure 30. Random forest feature importance for the transiting-anchoring binary classifier.

As shown in Figure 30, the most important features for classifying between transiting ships and anchored ships are the mean speed, standard deviation of speed, and accumulated angle. This does indeed make intuitive sense since anchored ships have a very low speed and do not turn while transiting vessels travel at faster speeds during their trip. Next, the Monte Carlo simulation (100 trials) classification results for several different model choices are shown in Table 6 for a comparative study of model choices.

Table 6. Random forest model summary for transiting-anchoring binary classifier.

| Number of Trees | PCA? | Standard Scaling? | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|---|
| 10 | No | No | 98.8% | 99.4% | 99.3% | 99.3% |
| 10 | No | Yes | 98.7% | 99.2% | 99.3% | 99.3% |
| 10 | Yes | No | 98.4% | 99.1% | 99.2% | 99.1% |
| 10 | Yes | Yes | 97.9% | 98.7% | 99.0% | 98.8% |
| 50 | No | No | 98.9% | 99.3% | 99.5% | 99.4% |
| 50 | No | Yes | 98.8% | 99.2% | 99.5% | 99.3% |
| 50 | Yes | No | 98.5% | 99.1% | 99.3% | 99.2% |
| 50 | Yes | Yes | 98.1% | 98.7% | 99.3% | 99.0% |
| 100 | No | No | 98.7% | 99.2% | 99.4% | 99.3% |
| 100 | No | Yes | 98.9% | 99.3% | 99.4% | 99.4% |
| 100 | Yes | No | 98.5% | 99.0% | 99.3% | 99.1% |
| 100 | Yes | Yes | 98.1% | 98.6% | 99.3% | 98.9% |

In this case, all results for all choices were extremely high performance likely due to the stark contrast between transiting and anchored. Such performance, as shown later, is uncharacteristic of most machine learning methods and should initially cause some alarm to a researcher. However, in this case, since transiting and anchoring are so different it is probable that these results are indicative of the contrast between the activities.

### 2. Fishing Binary Classifier

The second binary classifier built utilizes data from fishing vessels in order to classify ships that are either fishing or engaged in any other activity. This dataset has 240 data objects of which about 37.1% are vessels fishing and 62.9% are vessels engaged in any other activity. The features used for this classification task are: mean speed, standard deviation of speed, number of turns, normed number of turns, skew of speed, number of

speed changes, normed number of speed changes, number of stops, and accumulated angle. An example of feature importance is shown in Figure 31.

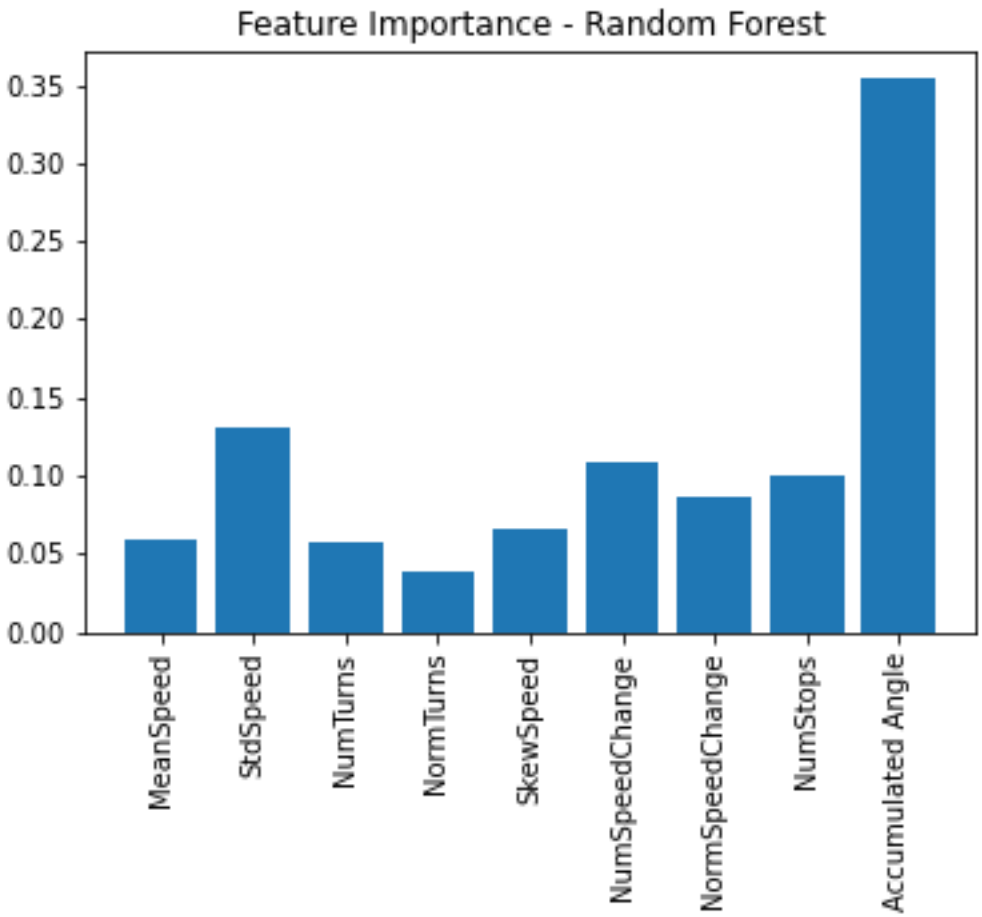



Figure 31. Random forest feature importance for the fishing binary classifier.

It is important to point out that in the behavior taxonomy portion of this thesis it was noted that fishing vessels often make many turns during their fishing activity. Thus, it should be no surprise that the accumulated angle feature, which accounts for turns, was the most important feature in the classification of fishing activity. Next the Monte Carlo simulation (100 trials) classification results for several different model choices are shown in Table 7 for a comparative study of model choices.

Table 7. Random forest model summary for the fishing binary classifier.

| Number of Trees | PCA? | Standard Scaling? | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|---|
| 10 | No | No | 89.2% | 85.6% | 84.8% | 84.9% |
| 10 | No | Yes | 89.2% | 85.8% | 85.3% | 85.2% |
| 10 | Yes | No | 90.5% | 88.3% | 85.9% | 86.8% |
| 10 | Yes | Yes | 89.8% | 88.0% | 84.2% | 85.7% |
| 50 | No | No | 90.0% | 85.2% | 88.6% | 86.6% |
| 50 | No | Yes | 90.3% | 84.9% | 89.7% | 87.0% |
| 50 | Yes | No | 91.4% | 86.9% | 90.4% | 88.3% |
| 50 | Yes | Yes | 90.5% | 86.9% | 87.6% | 87.0% |
| 100 | No | No | 89.7% | 84.8% | 88.6% | 86.4% |
| 100 | No | Yes | 90.0% | 84.5% | 88.9% | 86.4% |
| 100 | Yes | No | 92.1% | 89.2% | 90.3% | 89.5% |
| 100 | Yes | Yes | 91.4% | 88.4% | 88.7% | 88.3% |

For the results in Table 7, performance generally increased when the number of trees were increased. Additionally, performance typically improved when using PCA without any prior scaling. In fact, the highest performance of the classifier occurred with 100 trees with PCA enabled. As a result, the highest performance achieved was with a 92.1% accuracy, 89.2% precision, 90.3% recall, and 89.5% F1-score.

### 3. Ferrying Binary Classifier

The third binary classifier built utilizes data from passenger vessels in order to classify ships that are either ferrying or engaged in any other activity. This dataset has 237 data objects of which about 45.1% are vessels fishing and 54.9% are vessels engaged in any other activity. The features used for this classification task are: mean speed, standard deviation of speed, number of turns, normed number of turns, skew of speed, number of speed changes, normed number of speed changes, number of stops, and accumulated angle. An example of feature importance is shown in Figure 32.

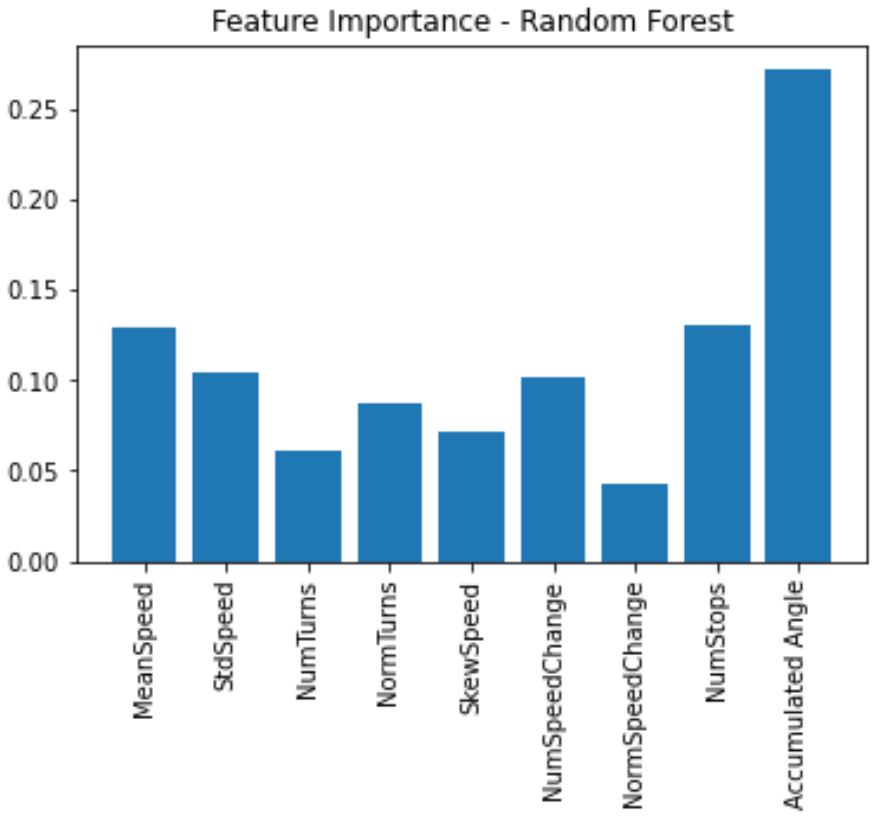



Figure 32. Random forest feature importance for the ferrying binary classifier.

In the behavior taxonomy portion of this thesis, it was noted that ferrying vessels often make many turns on a repetitive track. Therefore, it makes sense that the accumulated angle and number of stops were the most important features. Next, the Monte Carlo simulation (100 trials) classification results for several different model choices are shown in Table 8 for a comparative study of model choices.

Table 8. Random forest model summary for the ferrying binary classifier.

| **Number of Trees** | **PCA?** | **Standard Scaling?** | **Accuracy** | **Precision** | **Recall** | **F1** |
|---|---|---|---|---|---|---|
| 10 | No | No | 86.5% | 87.9% | 82.1% | 84.5% |
| 10 | No | Yes | 85.7% | 87.2% | 80.9% | 83.6% |
| 10 | Yes | No | 85.7% | 86.3% | 81.4% | 83.4% |
| 10 | Yes | Yes | 84.4% | 85.1% | 79.5% | 81.9% |

| Number of Trees | PCA? | Standard Scaling? | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|---|
| 50 | No | No | 86.8% | 86.8% | 83.5% | 84.8% |
| 50 | No | Yes | 86.5% | 86.2% | 84.4% | 85.0% |
| 50 | Yes | No | 87.1% | 85.2% | 86.1% | 85.4% |
| 50 | Yes | Yes | 85.7% | 85.1% | 83.1% | 83.8% |
| 100 | No | No | 87.1% | 86.4% | 85.2% | 85.6% |
| 100 | No | Yes | 87.1% | 86.4% | 85.1% | 85.5% |
| 100 | Yes | No | 87.5% | 86.6% | 85.5% | 85.8% |
| 100 | Yes | Yes | 86.2% | 85.2% | 83.9% | 84.2% |

Generally, for the results shown in Table 8, performance increased when the number of trees was increased. However, this time the PCA played less of a role in boosting performance. Regardless, the highest performance classifier (as judged by the F1-score) was the one with 100 trees and with PCA only enabled. This classifier had an accuracy of 87.5%, precision of 86.6%, recall of 85.5%, and F1-score of 85.8%.

### 4. Piloting Binary Classifier

The fourth binary classifier built utilizes data from pilot vessels in order to classify ships that are either piloting or engaged in any other activity. This dataset has 119 data objects of that about 52.9% are vessels piloting and 47.1% are vessels engaged in any other activity. The features used for this classification task are mean speed, standard deviation of speed, number of turns, normed number of turns, skew of speed, number of speed changes, normed number of speed changes, number of stops, and accumulated angle. An example of feature importance is shown in Figure 33.

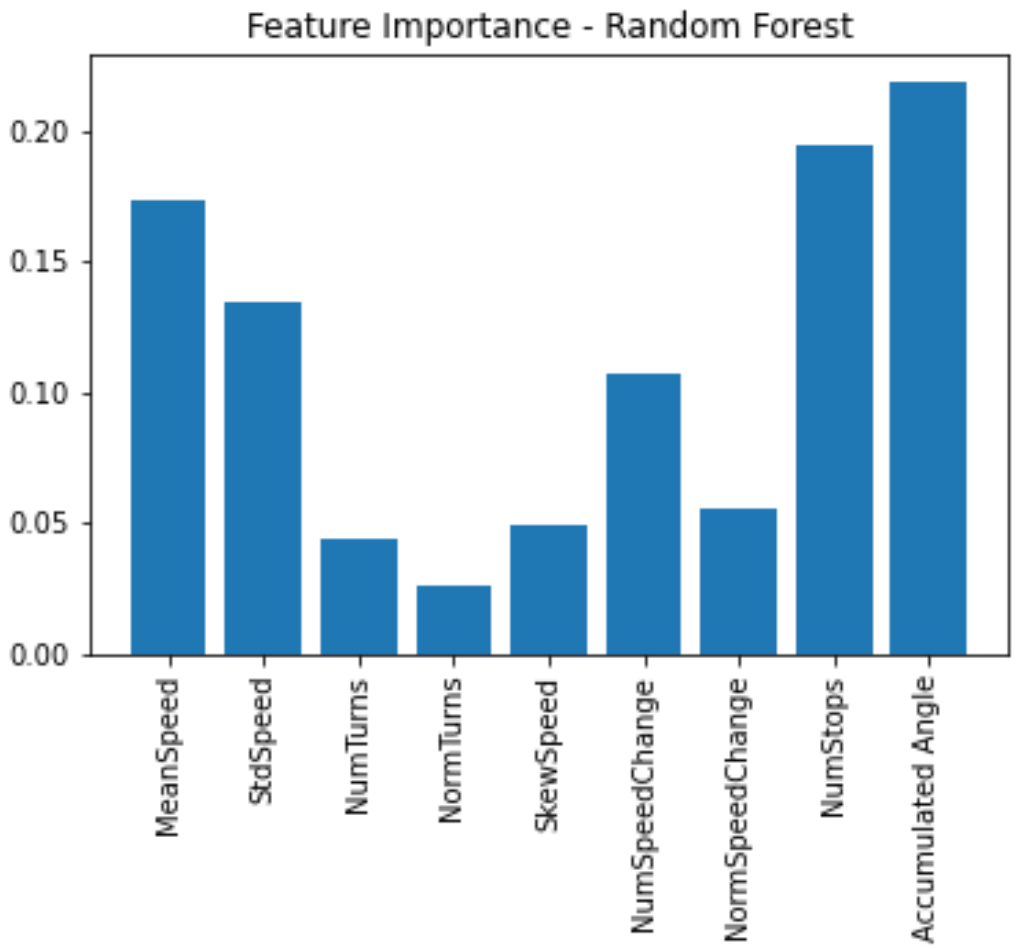



Figure 33. Random forest feature importance for the piloting binary classifier.

Recall that in Section III.B.6, the behavior taxonomy portion of this thesis, it was noted that piloting vessels often make many turns on a repetitive track with a common origin and many destinations. Therefore, it makes sense that the accumulated angle and number of stops were the most important features. This means that piloting does indeed look similar to ferrying as suggested through their model feature importance. Next, the Monte Carlo simulation (100 trials) classification results for several different model choices are shown in Table 9 for a comparative study of model choices.

Table 9. Random forest model summary for the piloting binary classifier.

| Number of Trees | PCA? | Standard Scaling? | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|---|
| 10 | No | No | 89.4% | 89.4% | 91.7% | 90.2% |
| 10 | No | Yes | 89.4% | 89.7% | 91.2% | 90.0% |
| 10 | Yes | No | 89.1% | 88.9% | 91.2% | 89.7% |
| 10 | Yes | Yes | 87.0% | 88.3% | 87.5% | 87.6% |
| 50 | No | No | 90.2% | 88.6% | 94.1% | 91.0% |
| 50 | No | Yes | 91.3% | 89.3% | 95.0% | 91.9% |
| 50 | Yes | No | 91.0% | 89.3% | 94.9% | 91.8% |
| 50 | Yes | Yes | 88.3% | 88.9% | 90.1% | 89.1% |
| 100 | No | No | 90.1% | 88.4% | 94.2% | 91.0% |
| 100 | No | Yes | 90.1% | 87.1% | 95.4% | 90.8% |
| 100 | Yes | No | 91.0% | 88.4% | 95.5% | 91.6% |
| 100 | Yes | Yes | 88.6% | 88.2% | 90.9% | 89.2% |

Generally, for the results shown in Table 9, performance increased when the number of trees were increased. However, this time the PCA played less of a role in boosting performance and provided a negligible difference to performance. The highest performance classifier (as judged by the F1-score) was the one with 50 trees and with standard scaling only enabled. This classifier had an accuracy of 91.3%, precision of 89.3%, recall of 95.0%, and F1-score of 91.9%. Thus, this shows how performance can start to saturate as the number of trees get larger—the model complexity continues to increase without any tangible benefit.

### 5. Multi-Class Classifier

Finally, a multi-class classifier was built utilizing data from all of the previous datasets in order to include all aforementioned classes: anchored, transiting, fishing, ferrying, and piloting. This dataset has 1383 data objects of which approximately 19.5% are vessels anchored, 61.7% are vessels transiting, 6.5% are vessels fishing, 4.6% are

vessels piloting, and 7.7% are vessels ferrying. The features used for this classification task are: mean speed, standard deviation of speed, number of turns, normed number of turns, skew of speed, number of speed changes, normed number of speed changes, number of stops, accumulated angle, and accumulated distance. The new feature of accumulated distance was added in order to help differentiate piloting/ferrying from transiting. An example of feature importance is shown in Figure 34.

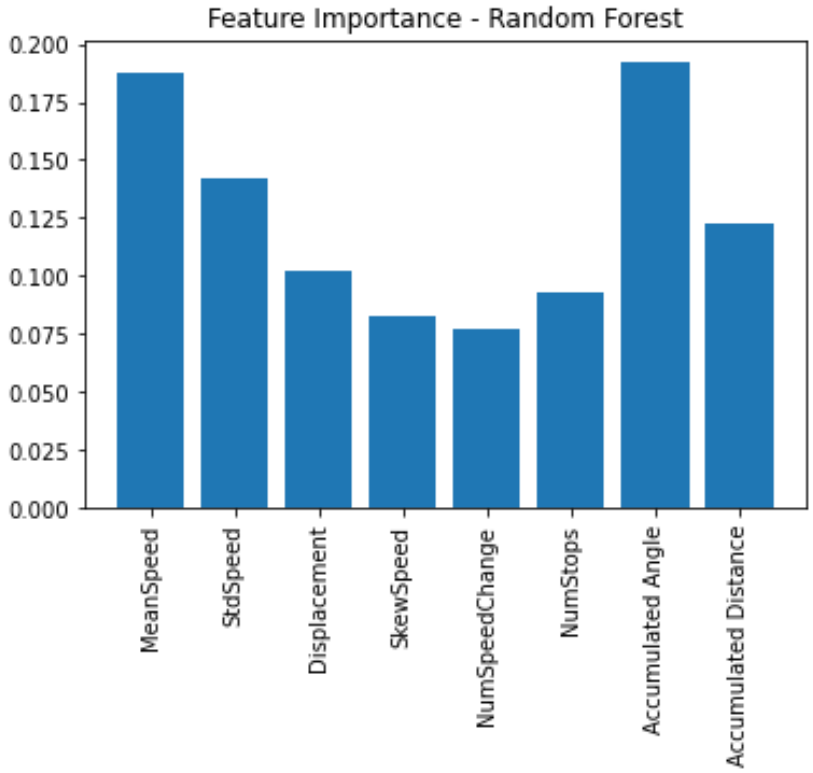



Figure 34. Random forest feature importance for the multi-class classifier.

Thus, from the feature importance graph in Figure 34, it is clear that the features that were identified as being important in the binary classifiers are also important in the context of the multi-class classifier. Therefore, if additional behaviors are desired to be classified in the future it will likely be helpful to first classify them in a binary classifier to identify pertinent features before implementing them in a multi-class classifier. It is also noted that the accumulated distance feature is also identified as an important feature for the

multi-class problem. Next the Monte Carlo simulation (200 trials) classification results for several different model choices are shown in Table 10 for a comparative study of model choices. Before looking at aggregate results, it is important to first look at an example of a confusion matrix to understand which behaviors are getting misidentified by the classifier. An example confusion matrix is shown in Figure 35.

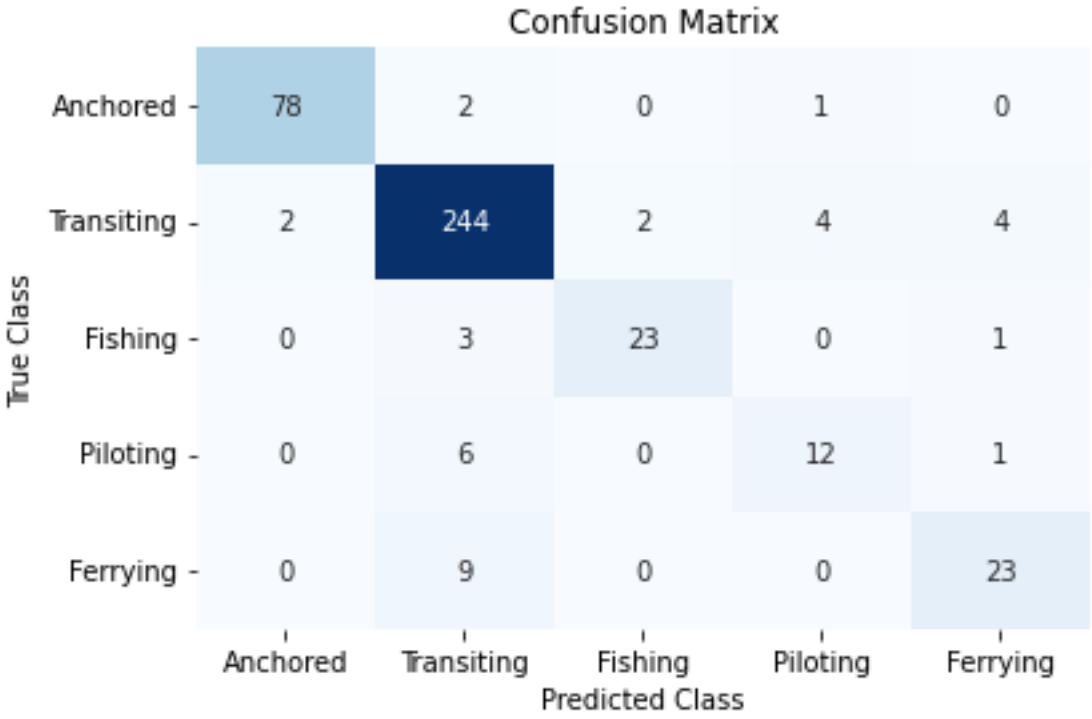



Figure 35. Confusion matrix for the multi-class classifier.

From the confusion matrix in Figure 35, it appears that transiting is the most misidentified class and is most confused with piloting and ferrying. In the context of a multi-class classifier, precision and recall take on slightly different meanings as well. Each class has its own precision, recall, and F1-scores. Therefore, to evaluate model performance, the mean results for each class are used. Classification results are shown in Table 10.

Table 10. Random forest model summary for the multi-class classifier.

| Number of Trees | PCA? | Standard Scaling? | Aggregate Accuracy | Aggregate Precision | Aggregate Recall | Aggregate F1 |
|---|---|---|---|---|---|---|
| 10 | No | No | 89.0% | 83.0% | 76.3% | 79.0% |
| 10 | No | Yes | 89.0% | 83.1% | 76.5% | 79.1% |
| 10 | Yes | No | 89.1% | 83.3% | 77.1% | 79.5% |
| 10 | Yes | Yes | 88.3% | 81.7% | 74.6% | 77.3% |
| 50 | No | No | 90.3% | 84.8% | 80.0% | 81.9% |
| 50 | No | Yes | 90.2% | 84.7% | 79.6% | 81.7% |
| 50 | Yes | No | 90.0% | 84.6% | 79.3% | 81.4% |
| 50 | Yes | Yes | 89.2% | 83.1% | 77.0% | 79.3% |
| 100 | No | No | 90.4% | 85.0% | 80.2% | 82.1% |
| 100 | No | Yes | 90.2% | 84.9% | 79.9% | 81.9% |
| 100 | Yes | No | 90.2% | 84.7% | 79.8% | 81.7% |
| 100 | Yes | Yes | 89.2% | 83.0% | 77.3% | 79.5% |

Generally, for the results shown in Table 10, performance increased when the number of trees were increased. PCA and scaling surprisingly both played less of a role in the performance. The highest performance classifier (as judged by the F1-score) was the one with 100 trees with only the original features. This classifier had an accuracy of 90.4%, precision of 85.0%, recall of 80.2%, and F1-score of 82.1%. Therefore, the multi-class classifier was relatively successful in distinguishing between five different classes.

# VI. CONCLUSIONS

The overarching goal and motivation of this thesis is to further develop MDA. This was accomplished by developing an end-to-end algorithm that ingests AIS data, filters it, extracts features, performs model selection and classifies the behavior of a maritime vessel. Additionally, this thesis develops an algorithm for the automatic identification of straight tracks in a region. This thesis has also established a comprehensive ship behavior taxonomy, which helps inform the intuition for the ship behavior classification models.

## A. SIGNIFICANT CONTRIBUTIONS

The significant contributions of this thesis are threefold: multiple ship behavior classifiers based upon features extracted from spatiotemporal data obtained through AIS, the development of an automatic track identifier, and the development of a comprehensive ship behavior taxonomy.

Multiple random forest classifiers were built for the identification of several ship behaviors. The random forest model is a particularly good choice for the enhancement of MDA due to their interpretability especially in regards to how random forest models identify the most important features for a given classification task. These classifiers were able to successfully classify between five different behaviors: anchored, transiting, fishing, ferrying, and piloting. Before developing a multi-class classifier, several binary classifiers were first developed that improved understanding of the maritime domain by identifying the key features for each ship behavior. In fact, the key features identified by each classifier also shed light on the behaviors themselves.

The key features identified for transiting and anchoring were mean speed, standard deviation of speed, and accumulated angle. This makes sense as perhaps the greatest difference between ships that are transiting or anchored is that ships that are transiting are moving.

The key features identified for the fishing binary classifier were accumulated angle, and standard deviation of speed. This makes sense according to the behavior taxonomy since fishing ships make many turns at changing speeds when engaged in fishing.

Therefore, the ship behavior taxonomy can inform the model and the model can inform the taxonomy.

The key features identified for the ferrying binary classifier were accumulated angle, number of stops, and mean speed. This makes sense according to the behavior taxonomy since passenger ships often travel over the same routes repetitively making many stops along the way. Therefore, the model indeed confirmed the intuitive knowledge found within the taxonomy.

The key features identified for the piloting binary classifier were accumulated angle, number of stops, and mean speed. This makes sense according to the behavior taxonomy since pilot ships would make many repetitive trips using a common origin to many possible destinations in the harbor. Thus, as identified in the taxonomy, there are many similarities between ferrying and piloting.

Finally, a multi-class classifier for all five activities was created. This classifier was successful in identifying the five different activities with (at best) an accuracy of 90.4%, a precision of 85.0%, a recall of 80.2%, and a F1-score of 82.1%. Thus, the random forest model was a good choice for the identification and classification of several ship behaviors. The results of this thesis can be used to help automate behavior classification tasks for an operator.

Next, a method to identify ship tracks using the Hough transform and DBSCAN clustering algorithm was developed. This algorithm is best summarized in Figure 25. As a result of this algorithm, the most populous tracks in the Baltic Sea were successfully identified. However, although this method was applied to ships traveling in the Baltic Sea, this method is generalizable to any area at any time. Therefore, this algorithm can be used to identify tracks and can also be used in a variety of spatiotemporal contexts. This method therefore has the potential to be used as an automatic identifier of new and developing sea lanes.

Additionally, through the application of the DBSCAN algorithm, anchorage areas were first predetermined and filtered out before identifying tracks in the Baltic Sea. This is because anchorage areas are characterized by extremely high dense areas of spatiotemporal

data. Therefore, the DBSCAN clustering algorithm is also able to be used in order to identify and characterize anchorage areas within AIS data through batch processing. For the purposes of track identification, these anchorage areas were filtered out after being identified by the DBSCAN algorithm. However future uses of this algorithm could certainly use this knowledge for other purposes and applications.

The ship behavior taxonomy, which until this point has been lacking from the literature, included information on multiple ship types including cargo ships, tanker ships, passenger ships, fishing ships, tugboats, and pilot ships. This taxonomy listed multiple behaviors and described them both analytically and visually with sample behavior patterns. The description of these behaviors is particularly useful in motivating feature selection and extraction for use in classification models.

As a note of caution—while machine learning and artificial intelligence offer many benefits towards automating the processing of big data—there are many pitfalls and dangers in the overreliance of this advancement in technology. As previously mentioned, machine learning models can be wrong when in important mission-critical situations, it is important to be right. Therefore, it is suggested that the results of this technology be used only to alert the operator for further processing and decision making. Management must also be wary that operators are not succumbing to personal biases and over relying on the output of machine learning models. Therefore, always keeping a human in the decision-making process can mitigate the risks of employing machine learning systems. Further analysis of the pitfalls of overreliance on machine learning technology is another area of future work that could be beneficial to the operator or decision maker.

Additionally, the entirety of this work was conducted using only spatiotemporal data contained within AIS data—i.e., only UTC timestamped positional data (latitude and longitude). The use of AIS for this thesis was chosen for its practicality and convenience. AIS data is, after all, widely available and possibly the only data operators may have available to them. However, although this thesis utilized AIS data, the models and methods contained herein are not limited nor constrained to AIS data and therefore can be expanded to many other data sources. Therefore, this thesis also supports the Center for Multi-INT Studies (CMIS) research group at NPS whose mission is to “expand the breadth and depth

of Multi-INT research, develop high-quality Multi-INT education programs, and facilitate and advance cohesion among Multi-INT practitioners from public, private and academic organizations."

The primary goal of this thesis was to develop MDA in order to ease the job of operators and decision makers. In this thesis, understanding of the maritime domain was enhanced through the development of a comprehensive ship behavior taxonomy, a method for automatic track identification, and several ship behavior classifiers. Therefore, this thesis stands with other works that have been helpful towards the development of MDA.

## B. FUTURE WORK

MDA is a broad field from which there are many avenues of discovery. Recommendations for future work regarding this thesis lie in two distinct categories—extensions to the classifiers previously built and the use of this classifier to inform other methods of MDA development.

One notable area of expansion is in the identification of other ship behaviors of interest. A non-exhaustive list of potential additional ship behaviors for classification are illegal activities, cooperative behaviors (such as towing, following, collision avoidance, etc.), and other stand-alone behaviors (trawling, long-lining fishing, sailing, etc.). As demonstrated in this thesis, the bulk of the work in classifying ship behaviors is in extracting expressive features from spatiotemporal data and applying these features to a machine learning model with a labeled data set. Therefore, the challenge in future work will be in the generation of labeled data sets for behaviors of interest. Representing behaviors that are infrequent may present additional issues. Therefore, it may be necessary to cooperate with other entities with repositories of expert-labeled AIS data in order to properly represent and classify new behaviors of interest.

While this work has focused primarily on identifying if a particular vessel is engaged in a particular activity or not, future work can instead be used to classify a geographical area as being tied to a particular behavior. For instance, is this area a fishing hole? Is this area used for transiting? For anchoring? Therefore, future work can focus upon generating a probabilistic model of the activities supported by a given region. It is possible

that future work could use the models and methods presented in this thesis to predict activities occurring within a geographical region to generate such a probabilistic model.

Many of these research topics are also of interest to many other groups and it is possible that future work will involve collaboration. Collaboration is particularly helpful in sharing datasets for a more informed analysis. It is also through this collaboration that fruitful results and ideas for future research can be generated. Therefore, there are many ways forward from this research in order to generate additional insight towards MDA.

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES


[1] Department of Homeland Security, "National plan to achieve maritime domain awareness," Oct. 2005 [Online]. Available: https://www.dhs.gov/xlibrary/assets/HSPD_MDAPlan.pdf

[2] International Maritime Organization, "AIS transponders." Accessed Oct. 1, 2021 [Online]. Available: https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx

[3] C. Campbell, "Torrent of print strains the fabric of libraries," *The New York Times,* Feb. 25, 1985 [Online]. Available: https://www.nytimes.com/1985/02/25/us/torrent-of-print-strains-the-fabric-of-libraries.html

[4] J. Manyika et al., "Big data: The next frontier for innovation, competition, and productivity," McKinsey Global Institute, New York, NY, USA, 2011 [Online]. Available: https://www.mckinsey.com/~/media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/big%20data%20the%20next%20frontier%20for%20innovation/mgi_big_data_full_report.pdf

[5] Domo, "Data never sleeps 9.0." Accessed Mar. 14, 2022 [Online]. Available: https://www.domo.com/learn/infographic/data-never-sleeps-9

[6] Spire Global, "How big is Spire AIS data." Accessed Mar. 14, 2022 [Online]. Available: https://faq.spire.com/how-big-is-spire-ais-data

[7] Brown Institute, "Thinking with computation," May 11, 2019 [Online]. Available: https://brown.columbia.edu/thinking-with-computation/

[8] V. Yee and J. Glanz, "How one of the world's biggest ships jammed the Suez Canal," *New York Times,* July 19, 2021 [Online]. Available: https://www.nytimes.com/2021/07/17/world/middleeast/suez-canal-stuck-ship-ever-given.html

[9] A. Saraiva and B. Murray, "Every step of the global supply chain is going wrong—all at once," Bloomberg, Nov. 22, 2021 [Online]. Available: https://www.bloomberg.com/graphics/2021-congestion-at-americas-busiest-port-strains-global-supply-chain/

[10] I. Kontopoulos, G. Spiliopoulos, D. Zissis, K. Chatzikokolakis and A. Artikis, "Countering real-time stream poisoning: an architecture for detecting vessel spoofing in streams of AIS data," in *IEEE 16th Intl. Conf. Dependable Auton. Secure Comput.*, 2018 [Online]. https://doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00139

[11] C.K. Pham, "Predicting the next port visit of a vessel using AIS data," M.S. thesis, Dept. of Oper. Res., NPS, Monterey, CA, USA, 2019 [Online]. Available: https://calhoun.nps.edu/handle/10945/64046

[12] S.P. Liraz, "Ships' trajectories prediction using recurrent neural networks based on AIS data,' M.S. thesis, Dept. of Oper. Res., NPS, Monterey, CA, USA, 2018 [Online]. Available: https://calhoun.nps.edu/handle/10945/60431

[13] I. Kontopoulos, K. Chatzikokolakis, K. Tserpes, and D. Zissis, "Classification of vessel activity in streaming data," in *Proc. 14th ACM Intl. Conf. DEBS,* 2020 [Online]. Available: https://doi.org/10.1145/3401025.3401763

[14] E.N. de Souza, K. Boerder, S. Matwin, and B. Worm, "Improving fishing pattern detection from satellite AIS using data mining and machine learning," *PLoS ONE*, vol. 11, no. 7, July 2016 [Online]. Available: https://doi.org/10.1371/journal.pone.0158248

[15] Spire Global, "Introduction to Automatic Identification Systems (AIS)." Accessed Apr. 26, 2022 [Online]. Available: https://spire.com/whitepaper/maritime/introduction-to-automatic-identification-systems-ais/

[16] Bae Systems, "What is multi-int signal processing?." Accessed Feb. 28, 2022 [Online]. Available: https://www.baesystems.com/en-us/definition/what-is-multi-int-signal-processing

[17] R.O. Duda and P.E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," Artif. Intell. Ctr., Menlo Park, CA, USA, Rep. 0704–0188, 1971 [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA457992.pdf

[18] K. Latt, "Sonar-based localization of mobile robots using the Hough transform," M.S. thesis, Dept. of Elec. and Comp. Eng., NPS, Monterey, CA, USA, 1997 [Online]. Available: https://calhoun.nps.edu/bitstream/handle/10945/8981/sonarbasedlocali00latt.pdf?sequence=1&isAllowed=y

[19] G.J. Bergues, C. Schurrer, and N. Brambilla, "Straight line detection through sub-pixel Hough transform," *Adv. Intell. Syst.,* vol. 998, pp. 1129–1137, July 2019 [Online]. Available: https://doi.org/10.1007/978-3-030-22868-2_75

[20] P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining,* 1st ed. Boston, MA, USA: Pearson Education, Inc., 2006.

[21] C. Albon, *Machine Learning with Python Cookbook,* 1st ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2018.

[22] A. Navlani, A. Fandango, and I. Idris, *Python Data Analysis,* 3rd ed. Birmingham, UK: Packt Publishing, Ltd., 2021.

[23] M.H. Assaf, V. Grouza, and E.M. Petriu, "The use of Kalman filter techniques for ship track estimation," *WSEAS Trans. on Syst.,* vol. 19, Feb. 2020 [Online]. Available: https://wseas.com/journals/systems/2020/a045102-060.pdf

[24] S. Narkhede, "Understanding confusion matrix," Towards Data Science, May 9, 2018 [Online]. Available: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

[25] A. Chakure, "Decision tree classification," Medium, July 5, 2019 [Online]. Available: https://medium.com/swlh/decision-tree-classification-de64fc4d5aac

[26] A.Y. Bhargava, *Grokking Algorithms—An Illustrated Guide for Programmers and Other Curious People*, 1st ed. Shelter Island, NY, USA: Manning, 2016.

[27] *Sci-kit Learn Decision Trees,* 1.0.2 ed., Sci-kit Learn, 2022 [Online]. Available: https://scikit-learn.org/stable/modules/tree.html

[28] R. Arif, "Classification in decision tree—a step by step CART (classification and regression tree)," Medium, Apr. 15, 2020 [Online]. Available: https://medium.com/analytics-vidhya/classification-in-decision-tree-a-step-by-step-cart-classification-and-regression-tree-8e5f5228b11e

[29] IBM, "Overfitting," Mar. 3, 2021 [Online]. Available: https://www.ibm.com/cloud/learn/overfitting

[30] L. Breiman, "Random forests," *Machine Learning,* vol. 45, Oct. 2001 [Online]. Available: https://doi.org/10.1023/A:1010933404324

[31] *Sci-kit Learn Feature Importances with a Forest of Trees,* 1.0.2 ed., Sci-kit Learn, 2022 [Online]. Available: https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html

[32] C.W. Therrien, *Discrete Random Signals and Statistical Signal Processing,* 1st ed. Englewood Cliffs, NJ, USA: Prentice-Hall, Inc., 1992.

[33] U.S. Coast Guard, *Automatic Identification System: USCG AIS encoding guide,* 2022 [Online]. Available: https://www.navcen.uscg.gov/pdf/AIS/AISGuide.pdf

[34] W. Niemela, F. Nicolas, M. Helavuori, and L. Meski, "HELCOM report on shipping accidents in the Baltic Sea 2019," HELCOM, Helsinki, FI, 2021 [Online]. Available: https://helcom.fi/wp-content/uploads/2021/12/HELCOM-report-on-Shipping-accidents-in-the-Baltic-Sea-2019-211207-FINAL.pdf

[35] A. Grimvall and K. Larsson, "Mapping shipping intensity and routes in the Baltic Sea," Swedish Institute for the Marine Environment, Gothenburg, SE, Rep. 2014:4, 2014 [Online]. Available: https://havsmiljoinstitutet.se/digitalAssets/1506/1506887_sime_ais_report_2014_5.pdf

[36] NOAA Office for Coastal Management, "Ports," Nov. 23, 2021 [Online]. Available: https://coast.noaa.gov/states/fast-facts/ports.html

[37] Baltic Lines, "Project findings 2019," Stavenger, NO, 2019 [Online]. Available: https://vasab.org/wp-content/uploads/2019/06/BalticLINes_project_findings_2019.pdf

[38] Mohit, "What are tanker ships?," Marine Insight, Dec. 30, 2021 [Online]. Available: https://www.marineinsight.com/types-of-ships/what-are-tanker-ships/

[39] Mohit, "What are passenger ships?," Marine Insight, Dec. 7, 2021 [Online]. Available: https://www.marineinsight.com/cruise/what-are-passenger-ships/

[40] Baltic Lines, "Project findings 2016," Stavenger, NO, 2016 [Online]. Available: https://vasab.org/wp-content/uploads/2018/06/Baltic-LINes-Shipping_Report-20122016.pdf

[41] C. Karan, "What are tug boats—types and uses," Marine Insight, Feb. 3, 2022 [Online]. Available: https://www.marineinsight.com/types-of-ships/what-are-tug-boats/

[42] Hiteshk, "What is a pilot boat?," Marine Insight, July 28, 2015 [Online]. Available: https://www.marineinsight.com/types-of-ships/what-is-a-pilot-boat/

[43] Nations Online, "Political map of the Baltic Sea." Accessed Feb. 28, 2022 [Online]. Available: https://www.nationsonline.org/oneworld/map/Baltic-Sea-map.htm

[44] H. Li, L. Chen, F. Li, and M. Huang, "Ship detection and tracking method for satellite video based on multiscale saliency and surrounding contrast analysis," *Appl. Remote Sens.,* vol. 13, no. 2, June 2019 [Online]. Available: https://doi.org/10.1117/1.JRS.13.026511

[45] W. Huang, B. Liu, X. Ding, and H. Zhang, "A ship detection and tracking method with time sequential shipborne radar imagery," in *Proc. SPIE,* vol. 7495, Oct. 2009 [Online]. Available: http://dx.doi.org/10.1117/12.831403

[46] N.R. Chopde and M.K. Nichat, "Landmark based shortest path detection by using A* and Haversine formula," *Intl. Innov. Res. Comp. Comm. Eng.,* vol. 1, no. 2, Apr. 2013 [Online]. Available: https://www.researchgate.net/publication/282314348_Landmark_based_shortest_path_detection_by_using_A_Algorithm_and_Haversine_Formula

[47] A. Upadhyay, "Formula to find bearing or heading angle between two points: latitude longitude," IGISMAP, 2015 [Online]. Available: https://www.igismap.com/formula-to-find-bearing-or-heading-angle-between-two-points-latitude-longitude/

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California