Applications of Machine Learning and Data Science to the Blue Economy: Sustainable Fishing and Weather Routing

by

Daniel Precioso Garcelán

A dissertation submitted by in partial fulfilment of the
requirements for the degree of Doctor of Philosophy in

Computer Science

UCA
Universidad
de Cádiz

Advisor:
David Gómez-Ullate Oteiza

Tutor:
Joaquín Pizarro Junquera

March 2023

To my family and friends.

ACKNOWLEDGEMENTS

This research was conducted in the Department of Computer Science at the Escuela Superior de Ingeniería, Universidad de Cádiz (UCA), Spain, between 2019 and 2023. This period has been an enriching experience, both intellectually and personally. I would like to express my gratitude to the people and organizations that have supported and guided me throughout this journey.

First and foremost, I want to express my heartfelt appreciation to my supervisor, Professor **David Gómez-Ullate**, whose unwavering support and guidance have been instrumental in the successful completion of this project. His patience, expertise, and invaluable advice have been a constant source of inspiration and motivation. I am deeply grateful for his friendship and mentorship, without which this study would not have been possible.

I am really thankful to Professor **Bernabé Dorronsoro**, leader of the Group of Optimization and Learning (GOAL), for inviting me to join his team. My heartfelt gratitude also goes to Professor **Joaquín Pizarro** for his guidance and support. I would also like to acknowledge Professor **Inmaculada Medina**, the head of UCASE, for granting me access to their laboratory, which was crucial to the success of this research.

Furthermore, I would like to thank Dr. **David Gordo**, Dr. **Alberto Torres**, and Dr. **Víctor Gallego** from Komorebi AI for their invaluable guidance and instruction, which made this industrial thesis possible. I am particularly grateful to Alberto for his exceptional leadership and for also serving as my industrial supervisor.

My heartfelt gratitude also goes to Professor **Robert Milson** for facilitating a research stay at the Department of Mathematics and Statistics of the Faculty of Science, Dalhousie University. His insightful comments and advice were instrumental in enhancing my research skills and capabilities.

I would also like to extend my appreciation to my colleagues and friends at UCA, Dalhousie University, and Komorebi AI for their collaboration, support, and friendship. They were all wonderful in their assistance, and I had a lovely time with them.

Thanks to Universidad de Cádiz, for the doctoral grant reference TDI-22-18 to undertake doctoral studies and for the financial support. Additionally, I would like to express my gratitude to the Department of Economy, Knowledge, Business and University of the Regional Government of Andalusia, for the funding received from project "Optimization and Machine Learning: A Two-Way Trip with Applications (OPTIMALE)" (reference FEDER-UCA18-108393).

I would also like to express my appreciation to Komorebi AI for providing me with the opportunity to collaborate with them and for their contract during my last year of research.

I am also grateful to the Association of Large Tuna Freezers (AGAC) for providing the logbook data used in the analysis of TUN-AI, and **Carlos Roa** for rendering available the Satlink echo-sounder dataset.

Finally, I would want to express my heartfelt appreciation to everyone who has contributed, directly or indirectly, to this four-year adventure. Thank you all for your support, encouragement, and inspiration!

Daniel Precioso Garcelán

Cádiz, Spain, 2023

SUMMARY

The Blue Economy encompasses an interdisciplinary field of study aimed at achieving sustainable utilization of ocean resources while preserving the environment's health. The importance of this concept lies in its role in achieving the Sustainable Development Goals defined by the United Nations. Nevertheless, the pursuit of economic development can often conflict with the principles of sustainability, underscoring the necessity of leveraging adequate tools to address these challenges.

Data science, and particularly Machine Learning, has become a valuable tool for addressing the challenges of the Blue Economy. For example, in the field of sustainable fishing, monitoring fish populations is highly relevant and can be achieved through Machine Learning models. In another area, such as maritime transport, the implementation of weather routing tools can optimize sea routes, improving fuel efficiency and ensuring a reduction in greenhouse gas emissions.

This thesis will delve into the study of sustainable fishing and weather routing in the context of the Blue Economy, applying data science techniques to improve efficiency and sustainability in both fields.

Keywords: Data Science, Machine Learning, Blue Economy, Sustainable fishing, Echo-sounder buoys, Weather routing.

RESUMEN

La Economía Azul ha surgido como un campo de estudio interdisciplinario que busca aprovechar los recursos del océano de manera sostenible y preservar su salud ambiental. Este concepto se ha vuelto cada vez más importante para alcanzar los Objetivos de Desarrollo Sostenible de las Naciones Unidas. Sin embargo, el desarrollo económico y la sostenibilidad pueden entrar en conflicto, lo que destaca la necesidad de abordar estos desafíos con herramientas adecuadas.

La ciencia de datos, y en particular el aprendizaje automático (Machine Learning), se ha convertido en una herramienta valiosa para abordar los desafíos de la Economía Azul. Por ejemplo, en el ámbito de la pesca sostenible, es muy relevante la monitorización de poblaciones de peces, que se puede realizar mediante modelos de Machine Learning. En otro ámbito, como es el transporte marítimo, la implementación de herramientas de “weather routing” puede optimizar las rutas por mar, mejorando la eficiencia en el consumo de combustible y garantizando una reducción en las emisiones de gases de efecto invernadero.

En esta tesis se profundizará en el estudio de la pesca sostenible y el weather routing en el contexto de la Economía Azul, aplicando técnicas de ciencia de datos para mejorar la eficiencia y sostenibilidad en ambos campos.

Palabras clave: Ciencia de Datos, Machine Learning,Economía Azul, Pesca sostenible, Boyas con ecosonda, Weather routing.

PUBLISHED AND SUBMITTED CONTENT

The following published articles have been (co)authored by me:

- D. Precioso and D. Gómez-Ullate, 'Sustainable Fishing: applying Data Science to the Ecological Trap Hypothesis,' *Actas de las Jornadas de Investigación Predoctoral en Ingeniería Informática*, p. 21, 2021
  - This work, presented during the first predoctoral consortium in Computer Science (JIPII 2021) organized by the University on Cádiz, was the introduction to my research on sustainable fishing.
  - As part of my industrial Phd, the study was supported by Komorebi AI, Satlink, OPAGAC and the Spanish Institute of Oceanography.
  - Chapter 2 expands on this work.
- D. Precioso, M. Navarro-García, K. Gavira-O'Neill *et al.*, 'TUN-AI: Tuna biomass estimation with Machine Learning models trained on oceanography and echosounder FAD data,' *Fisheries Research*, vol. 250, no. February, p. 106 263, 2022. DOI: `10.1016/j.fishres.2022.106263`
  - This study was done in collaboration with Komorebi AI and Satlink.
  - In this work we combine oceanographic data, echo-sounder buoy data and logs from fishing boats to evaluate different Machine Learning models and establish a pipeline, named TUN-AI , for processing echo-sounder buoy data and estimating tuna biomass.
  - My role was to analyse and preprocess the raw data provided by Satlink. I also contributed in building the pipeline to train the Machine Learning models with the aforementioned data.
  - Chapter 2 is based on this article. The material from this source included in this thesis is not singled out with typographic means and references.
- D. Gómez-Ullate, F. Amor, J. J. de la Jara *et al.*, 'Smart Shipping: Optimización de rutas marítimas en tiempo real,' *61 Congreso de Ingeniería Naval e Industria Marítima*, 2022
  - This work presented a weather routing system that uses real time data and forecasts into the future of navigation conditions to find the best possible route between two ports.
  - Data on ocean currents, wind and waves are obtained from prediction models from data providers like National Oceanic and Atmospheric Administration (NOAA) and Copernicus. The optimization algorithm was based on graph search, and used simple fuel consumption models as cost function.

    - My role was to preprocess the oceanography data and assist in the developing of the graph search algorithms.
    - Chapter 4 is based partially on this article. The material from this source included in this thesis is not singled out with typographic means and references.

The following articles have been also (co)authored by me and have been submitted for publication:

- M. Navarro-García, D. Precioso, K. Gavira-O'Neill *et al.*, 'How do tuna schools associate to dFADs? A study using echo-sounder buoys to identify global patterns,' *arXiv preprint arXiv:2207.07049*, 2022
    - This study was done in collaboration with Komorebi AI and Satlink.
    - The work is a continuation of the first article mentioned in this section [2]. Here we used TUN-AI to examine the temporal trends of tuna schools' association to drifting objects.
    - My role was to preprocess echo-sounder buoy data and apply Machine Learning models to predict tuna presence under those buoys. I then use these predictions to derive specific metrics that characterized tuna behaviour, and studied the difference of these metrics between oceans.
    - Chapter 3 is based on this article. The material from this source included in this thesis is not singled out with typographic means and references.
- D. Precioso, R. Milson, L. Bu *et al.*, 'Hybrid Search method for Zermelo's navigation problem,' 2023
    - This study was conducted during my research internship at Dalhousie University, Canada.
    - In this work we apply mathematical variational methods to solve the Zermelo navigation problem, both in synthetic vector fields and on real data (ocean currents).
    - My role was to adapt the mathematical equations to Python, and develop the benchmarks by building synthetic vector fields and preprocessing the real oceanography data.
    - Chapter 4 is based partially on this article. The material from this source included in this thesis is not singled out with typographic means and references.

OTHER RESEARCH MERITS

During my four years of research on Data Science and Machine Learning, I was involved in some other projects, unrelated to the Blue Economy but relevant to other industries. Below I list the articles produced during those researches:

- D. Precioso and D. Gómez-Ullate, 'Thresholding Methods in Non-Intrusive Load Monitoring,' *The Journal of Supercomputing*, 2023
  - Non-Intrusive Load Monitoring (NILM) aims to predict the consumption or status of domestic appliances in a household only by knowing the aggregated power load. NILM can be formulated as regression problem or most often as a classification problem. Most datasets gathered by smart meters allow to define naturally a regression problem, but the corresponding classification problem is a derived one, since it requires a conversion from the power signal to the status of each device by a thresholding method.
  - In this study we treated three different thresholding methods to perform NILM. We analysed the performance of Deep Learning state-of-the-art architectures on both the regression and classification problems, introducing criteria to select the most convenient thresholding method.
  - We concluded that a discussion of what is the most appropriate method should not be based on the performance achieved by estimation models alone, but include also some objective way to judge the interpretability of the results. We suggested as an objective criterion to use the intrinsic error, i.e. Mean Average Error between the original power series and the reconstructed binary series from the Deep Learning model.
  - This study was conducted fully by my Phd supervisor, David Gómez-Ullate, and I. My roles involved preprocessing the data, building the Machine Learning models, designing the scores to evaluate their predictions, and finally training those models and evaluating them.
- D. Precioso and D. Gómez-Ullate, 'Non-Intrusive Load Monitoring using Multi-Output CNNs,' in *2021 IEEE Madrid PowerTech*, 2021, pp. 1–6. DOI: `10.1109/PowerTech46648.2021.9494943`
  - This article also derived from our research on NILM, and was presented to the PowerTech conference, hosted in Madrid by IEEE.
  - In this paper we propose a modification of a state-of-the-art convolutional neural network architecture to allow for multi-output channels, solving the regression and classification problems with relative weights simultaneously.

As a reminder, the problems involve estimating the appliance power load and detecting ON/OFF states, respectively. We analyze the performance of this multi-output model and study the interplay between the two approaches on NILM.

- I. Barbeito, D. Precioso, M. J. Sierra *et al.*, 'Effectiveness of non-pharmaceutical interventions in nine fields of activity to decrease SARS-CoV-2 transmission (Spain, September 2020-May 2021),' *Frontiers in Public Health*, 2023
  - The project was a collaboration between University of A Coruña, Universidad de Cádiz and the Study Group for Non-Pharmaceutical Interventions (NPI) in Spain.
  - We estimated the association between the level of restriction in nine different fields of activity and SARS-CoV-2 transmissibility in Spain, dating from 15th September 2020 to 9th May 2021.
  - A stringency index was created for mobility, social distancing, commerce, indoor and outdoor bars and restaurants, culture and leisure, worship and ceremonies, indoor and outdoor sports, for each Spanish province daily. The logarithmic return of the weekly percentage variation of the 7-days COVID-19 cumulative incidence was used to measure COVID-19 transmission, lagged 12 days behind the stringency index. A hierarchical multiplicative model was fitted, and the median of coefficients across provinces was used to quantify the effect of increasing one standard deviation in the stringency index in each field.
  - Our results showed that highest levels of restriction were seen in mobility, sports and restaurants, particularly indoors. The increase in restrictions overall reduced SARS-CoV-2 transmission by 22% in one week.
  - My role in this study was to preprocess the data from NPI and compute the stringency index for each field of activity.
- A. Ruiz-Zafra, D. Precioso, B. Salvador *et al.*, 'NeoCam: An edge-cloud platform for non-invasive real-time monitoring in neonatal intensive care units,' *IEEE Journal of Biomedical and Health Informatics*, pp. 1–12, 2023. DOI: `10.1109/JBHI.2023.3240245`
  - NeoCam project was born at Universidad de Cádiz, in collaboration with Hospital Universitario Puerta del Mar (Cádiz, Spain).
  - We used an specialized device (the Luxonis OAK-D smart camera) to build a contactless monitoring system for newborn babies, specifically the ones at Intensive Care Units (NICUs). The NeoCam system was able to monitor pain and emotional stress, breathing rhythm, physical activity and sleep-wake cycles purely using Computer Vision and Machine Learning (ML) algorithms, which were developed specifically for this project.

  - All the vitals monitored by NeoCam are sent on real time to a server, and can be accessed by the personal at NICUs. The system also send alerts when it detects anomalies.
  - My role was to develop, train and implement the aforementioned ML algorithms.

Additionally, some of the projects I was involved in won international competitions:

- **OpenCV AI Competition 2021**.
  - OpenCV AI competition is focused on solutions solving real world problems using spatial AI. Teams use the new OpenCV AI Kit D (OAK-D) to solve their challenge areas. The OAK-D is a smart camera with neural inference and depth processing capability on board.
  - My team "Caleta", from Universidad de Cádiz, presented the project NeoCam. Refer to the paper described previously in this section.
  - Out of the 1400 submissions, NeoCam earned 2nd place in the International Final, and 1st place in the Europe + Russia + Australasia region.
  - Link to the official page of the competition: https://opencv.org/opencv-ai-competition-2021/
- **Ocean Hackathon 2021**.
  - Ocean Hackathon is an international event, organized by Campus Mondiale de la Mer. More than a hundred teams from 15 different cities presented their projects in this edition. The best projects of each participating city were invited to compete in the grand finale, hosted in Brest (France).
  - I was part of the project Smart Shipping. We represented the city of Cádiz during the international final and earned second place. This project later motivated the study shown in Chapter 4.
  - Link to the news kit: https://www.campusmer.fr/files/4189/OH6_-_press_kit_-_2022-02.pdf

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

aCAT Aggregation’s Continuous Absence Time

aCRT Aggregation’s Continuous Residence Time

AT Aggregation Time

AUC Area Under the Curve

BAI Buoy-Derived Abundance Index

CAT Continuous Absence Time

CMEMS Copernicus Marine Environment Monitoring Service

CPUE Catch Per Unit Effort

CRT Continuous Residence Time

CT Colonization Time

DFAD Drifting Fish Aggregating Device

DL Deep Learning

DS Data Science

DT Dissaggregation Time

EEDI Energy Efficiency Design Index

ENet Elastic Net

FAD Fish Aggregating Device

FDA Ferrero-de Diego-Almagro Algorithm

GA Genetic Algorithm

GAM Generalized Additive Model

GB Gradient Boosting

GHG Green House Gases

GLM Generalized Linear Model

HS Hybrid Search

HyCOM Hybrid Coordinate Ocean Model

IMO International Maritime Organization

IQR Inter-Quartile Range

LR Logistic Regression

MAE Mean absolute error

ML Machine Learning

MCR Maximum Continuous Rating

MLP Multi-Layered Perceptron

NOAA National Oceanic and Atmospheric Administration

OGCM Ocean General Circulation Model

OR Occupancy Rate

RF Random Forest

RK4 Fourth order Runge-Kutta method

ROC Receiver Operating Characteristic

SD Standard Deviation

SDG Sustainable Development Goals

SFOC Specific Fuel Oil Consumption

SL Supervised Learning

SMAPE Symmetric mean absolute percentage error

SSHa Sea Surface Height anomaly

ST Soak Time

SVM Support Vector Machine

UN United Nations

WASP Wind Assisted Ship Propulsion System

XGB XGBoost

ZIVP Zermelo Initial Value Problem

ZNP Zermelo's Navigation Problem

1. INTRODUCTION

The United Nations (UN) first used the term “Blue Economy” in 2012 [10], to describe the sustainable use of ocean resources [11] for economic growth, improved livelihoods, and jobs, while preserving the health of ocean ecosystem. It encompasses a wide range of activities, including fisheries, tourism, marine renewable energy, and transportation, among others.

The Blue Economy is closely linked to the concept of sustainable development , and specifically to the UN Sustainable Development Goals (SDG), which are a set of 17 global goals adopted by the UN General Assembly in 2015 to end poverty, protect the planet, and ensure peace and prosperity for all [12]. One of the key SDGs is SDG 14, which is specifically focused on conserving and sustainably using the oceans, seas, and marine resources for sustainable development [13]. This goal aims to prevent and significantly reduce marine pollution of all kinds, in particular from land-based activities, including marine debris and nutrient pollution. It also aims to protect and restore marine and coastal ecosystems, enhance their contributions to carbon sequestration and storage. By promoting the sustainable use of ocean resources, the Blue Economy can help countries achieve SDG 14 and other related SDGs, such as those on climate action, sustainable cities and communities, responsible consumption and production.

However, promoting the maritime industry while ensuring a sustainable use of the ocean resources is a complex task to achieve. It requires a deep understanding of the industrial activities to make them more efficient and environmentally friendly [14], [15]. One way to reach this goal is through Data Science (DS) implementations. The focus of DS is to extract knowledge from specific fields, then apply those insights to solve problems related to the field. ML is a specific application of DS, where complex models are trained with huge amounts of data to optimize specific tasks. In recent years, many successful applications of DS and ML to the Blue Economy have spurred [16]–[18].

This thesis presents two projects, both related to different fields inside the Blue Economy. In these projects ML and DS had been applied to improve the efficiency of maritime industrial tasks while also ensuring a sustainable development. The focus of this research is on practical applications, with a particular emphasis on addressing industry needs and challenges. By demonstrating the value of Data Science and Machine Learning for the Blue Economy, this thesis aims to contribute to the development of a more sustainable and inclusive ocean economy.

This chapter will briefly describe the basics of DS and ML, then introduce the core concepts related to each project: sustainable fishing and weather routing. In addition, it will outline the objectives pursued in this thesis and explain the general structure of the manuscript.

1.1. Data Science and Machine Learning

Data Science (DS) is a field of study that focuses on extracting knowledge from vast amounts of data [19]. It is an interdisciplinary field that combines elements of Computer Science, Statistics, and Mathematics to uncover patterns and insights in data.

The concept of DS originated in the early 1960s as a new discipline that would support the understanding and interpretation of the large amounts of data that were being generated at the time [20]. The first description of the field is attributed to Tukey [21] "(...) as I have watched mathematical statistics evolve, I have had cause to wonder and to doubt (...) I have come to feel that my central interest is in data analysis (...)", referring to the merging of statistics and computers. The term "Data Science" was coined in the 1970s as an alternative name for Computer Science [22]. In the years since, DS has evolved to include Statistics and Mathematics, in addition to Computer Science.

Today, DS is an important part of business and academic research [23]. Data scientists use modern tools and techniques to find hidden patterns in data, derive meaningful insights, optimize tasks, and make business decisions. They work with large datasets and use advanced algorithms and Machine Learning techniques to uncover insights that would not be possible using traditional methods.

DS has become increasingly important as businesses and organizations generate more and more data. The ability to extract insights from this data can provide a significant competitive advantage, enabling organizations to make better decisions and improve their operations. As a result, demand for data scientists with the skills and expertise to analyze and interpret data has grown rapidly in recent years.

As previously mentioned, one of the tools applied in Data Science is **Machine Learning (ML)**. The term "Machine Learning" was coined prior to DS, in the late 1950s [24] along with the synonym "self-teaching computers" [25]. It describes any computer algorithm that tries to emulate human behaviour in a specific task, using data and some evaluation method to improve its performance.

There are three major components of a ML system: data, models, and learning [26]. The data used by a ML system must be in numerical format, as computer programs work with tensors. Real data sets contain information in varied formats (such as text and images) and is usually the role of DS to preprocess these data and deliver it to the model in numeric format. A ML model *learns* from the data set and should be able to perform well on unseen data [27]. Learning can be understood as a way to automatically find patterns and structure in data by optimizing the parameters of the model [28]. Machine Learning is a broad field of study with many branches. This thesis will show applications of Supervised Learning and Deep Learning. Both categories are described below.

Supervised Learning (SL) is a subcategory of ML where each data point in the data set is labelled [29], meaning that the data includes both input and output examples. This allows the model to learn to map the input data to the corresponding output labels

[30]. In other words, the model is trained on a labelled dataset, and it learns to make predictions based on the relationships between the input data and the corresponding labels, see Figure 1.1. The two projects present in this manuscript, TUN-AI and weather routing, involve supervised tasks.

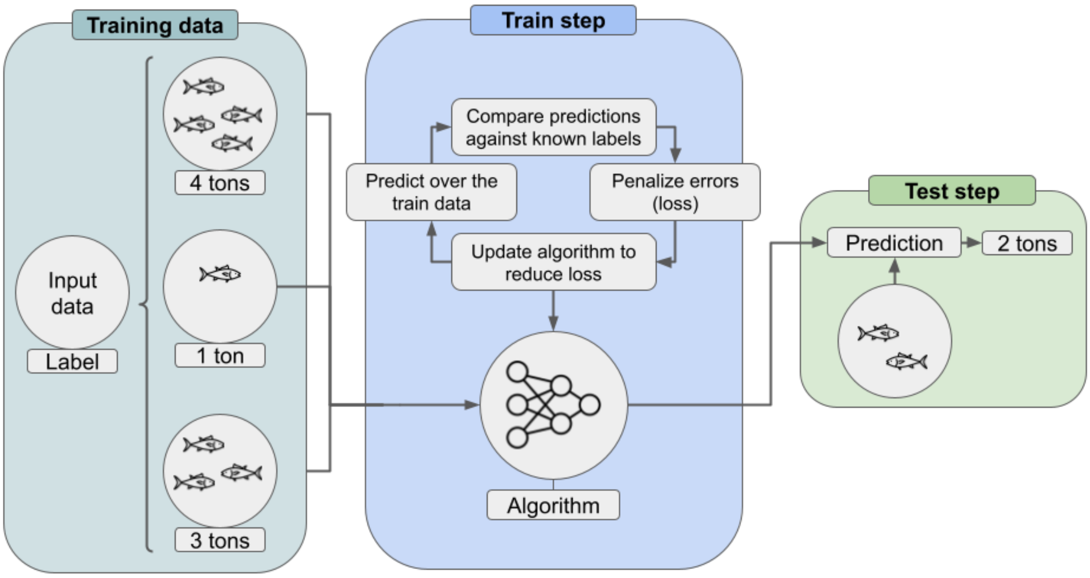


Figure 1.1: Scheme of how to develop a supervised learning algorithm.

Deep Learning (DL) is another subfield of ML that focuses on building dense models that are able to discover intricate structures in high-dimensional data and solve complex problems [31]. These models are often implemented using neural networks, which are composed of many interconnected processing nodes that are able to learn from data and make predictions or decisions based on the information they have learned. DL models are typically able to handle large datasets, and are able to learn complex patterns and relationships within the data.

1.2. Sustainable fishing

A conventional idea of a sustainable fishery is that it is one that is harvested at a sustainable rate, where the fish population does not decline over time because of fishing practices [32], [33]. This not only involves avoiding over-fishing but also ensuring that the methods used to locate and attract fish schools are not harmful for the overall fish population. Sustainability in fisheries requires a deep understanding of the fish's population dynamics in order to perform a correct stock assessment, which can be achieved by two different approaches: individual fish tagging and school detection.

Individual tagging involves attaching a device, such as a tag (see Figure 1.2), to a single fish and monitoring it over time through recaptures by the fishery. The goal of these studies is to create a group of tagged individuals that can be used to estimate mortality rates and population size, provided certain assumptions are met [34]. These assumptions include: that tagged fish experience the same processes and behave in the same way as untagged fish; that the probability of capturing a tagged fish is the same as that for an untagged fish in the specific area and time period being studied; and that tag losses due to shedding, tagging-induced mortality, and non-reporting of recaptured tagged fish are minimal or can be separately calculated [35], [36]. To meet these assumptions, experimental designs and/or analytical corrections may be used, such as releasing tags widely in the area being studied. Size-structured (or age-structured as a proxy for size) modelling approaches [36] can also be employed to address any discrepancies in the size distribution of tagged fish compared to the catch of the fishery. When these assumptions are satisfied, tagging data can provide valuable information for stock assessment, either through stand-alone analyses [37] or by being incorporated into the stock assessment model with other data [38].

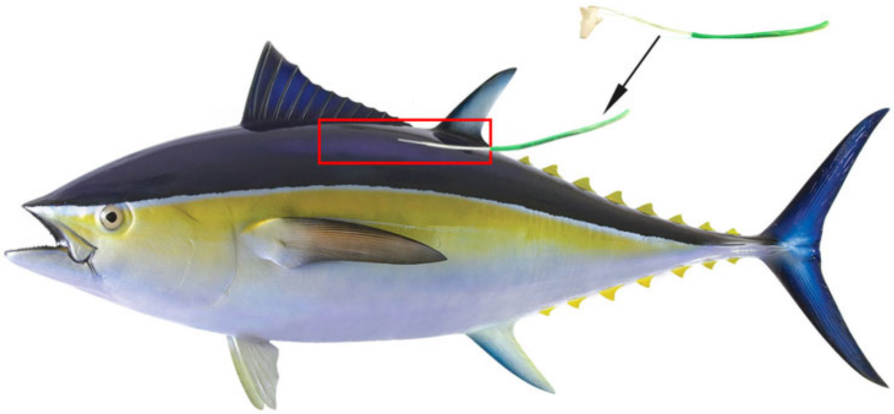


Figure 1.2: Fish tag placement. An unique ID is shown on each tag, that helps identify the individual. After a tag is placed, the fish is registered in a database and released into the ocean. When the fish is caught again, fishermen are encouraged to register where and when the fish was found, using the ID. Image source: `grayfishtagresearch.org`

The second approach for stock assessment is to **identify fish schools**, which requires the use of more sophisticated tools, such as acoustic detection devices (e.g. echo-sounder buoys). These instruments can cover a large area and provide real-time information on the distribution and behaviour of fish schools [39]. One advantage of using echo-sounder buoys is that they do not require the physical handling or tagging of individual fish, which can be time-consuming and potentially harmful to the specimen. This makes them a more humane and less invasive method for studying fish populations. Echo-sounder buoys are also relatively low-cost and easy to deploy [40]. However, studying the data collected by these devices requires sophisticated techniques. In this thesis, we study the possibility of applying DS, which is done in two steps. First, ML models are developed to estimate the size of fish schools (i.e. their biomass) in real time, using data from the detection devices and any other available sources (e.g. oceanography). Once the models are ready, they can be applied to estimate the size of fish populations across different regions, study their dynamics and ultimately give a stock assessment. It is important to note that, in order for the ML models to perform well, it is better to train them to identify a particular species. This thesis will focus on tuna.

To better understand the practices on tuna fishing and how echo-sounder buoys are used, it is necessary to introduce the concept of Fish Aggregating Device (FAD). These floating objects are designed and strategically placed to attract pelagic fish such as tuna [41], see Figure 1.3. This technique has been used by fishermen for centuries to aggregate fish species, throughout tropical and sub-tropical oceans [42]–[45]. When deployed in open ocean, a FAD is often called Drifting Fish Aggregating Device (DFAD) because it is not anchored to the seafloor. In tuna purse-seine fisheries, DFADs are commonly used to locate schools of skipjack tuna (*Katsuwonus pelamis*), yellowfin tuna (*Thunnus albacares*) and bigeye tuna (*T. obesus*), and are essential for increasing fishing efficiency. Today, more than 55% of tropical tuna caught by industrial purse-seine vessels in the Indian, Atlantic and Pacific oceans is caught using DFADs, accounting for 36% of the world's total tropical tuna catch [46]–[48].

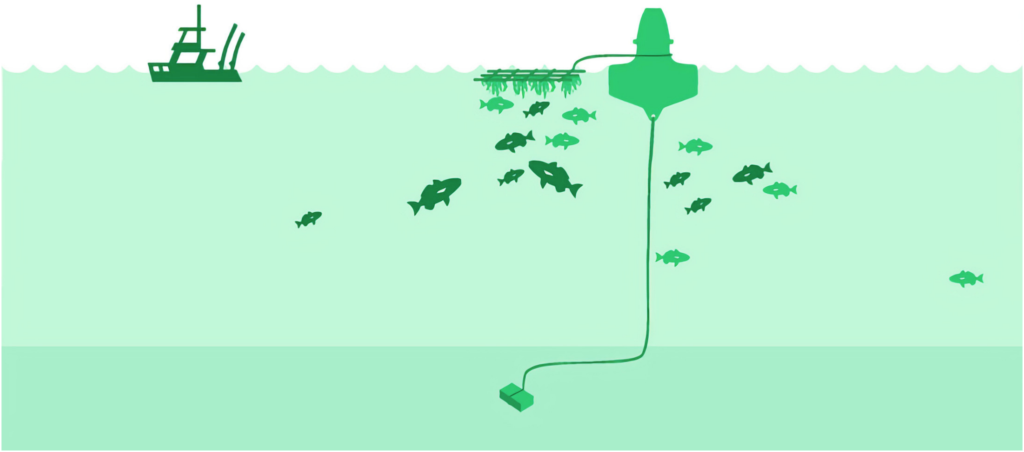


Figure 1.3: Man-made FAD anchored to the sea floor. Ropes and lines encourage the settlement of marine plants and small crustaceans and molluscs, which in turn attract small fish. Fish finders may be attached to a FAD allowing fishermen to electronically "connect" to the FAD and see how many and at what depth the fish are located. Image from NOAA [41].

The first observations that floating objects attract a variety of fishes were very early [49]. As long ago as 200AD, the Roman author Oppian, writing about fishing in the Mediterranean, recorded [50]: "*The fishermen gather reeds and tie them together in bundles which they let down into the waves and underneath they tie a heavy stone by way of ballast. All this they let sway gently in the water; and straight-way the shade-loving tribes of the Hippurus gather in shoals and linger about delightedly rubbing their backs against the reeds. Then the fishers row to them to find a ready prey, and bait their hooks and cast them, and the fish seize them, hastening therewith their own destruction.*"

Initially, DFADs were of natural origin [51], such as wooden debris or mammal carcasses, that fishermen would come across while searching for free-swimming schools of tuna. There were also DFADs unintentionally produced by human activities, such as the pieces of fishing nets [52]. In the 1960s, purse seine fishing boats began to fish on schools associated with natural floating objects [53].

In the mid-1980s, tools were developed to track DFADs using radar reflectors or radio. Later, GPS buoys were added to allow for remote location of the DFADs [40], [54]. The use of tracking buoys has been considered a significant technological development for increasing the efficiency of DFAD tuna fishing [55]. Most DFADs today are equipped with satellite-linked instrumented buoys, which include GPS and an echo-sounder, providing fishermen with accurate geolocation information and an estimate of the associated tuna biomass. This allows fishing crews to monitor their DFADs and the size of tuna aggregations in real-time, allowing them to target those with larger aggregated schools and increase their catch while reducing search effort [40], [56].

1.2.1. Tuna biomass estimation

The introduction of DFADs has resulted in major modifications in industrial purse-seine fishing fleets targeting tropical tunas, including adjustments to traditional measures used to determine Catch Per Unit Effort (CPUE), such as search-time and time-at-sea, [43]. Some studies have stressed the need of fishery-independent abundancy indices and the use of non-traditional data sources to track tuna stock health and the consequences of fishing pressure over time [57]–[59]. Echo-sounder buoys tied to DFADs across the world provide geo-referenced biomass estimations on a regular basis. Given the abundance and global dispersion of DFADs, the data supplied by these echo-sounder buoys might be useful. One example is the Buoy-Derived Abundance Index (BAI) for tropical tunas, which was reported by Santiago, Lopez, Moreno *et al.* [58] and is based on biomass estimations provided by three echo-sounder buoy brands in the Atlantic, Indian, and Pacific Oceans.

However, multiple studies have observed cases in which the biomass estimations produced by echo-sounder buoys varied considerably from the actual tuna tonnage collected by boats [60]–[62]. This might be attributed to the fluctuating nature of DFADs fish aggregations, which are frequently formed up of pelagic species other than tuna [44] and hence not included in vessel capture statistics, but would be counted in biomass estimates generated by echo-sounder buoys. Furthermore, oceanic conditions can influence fish distribution and behaviour, resulting in tuna aggregation patterns near DFADs [63]–[65]. It is critical to evaluate and understand the influence of these factors on the biomass estimates produced by these buoys in order to create a representative index of abundance using echo-sounder buoy data.

1.2.2. Tuna dynamics

Because of the extensive use of DFADs in fishing operations, a greater knowledge of the possible ecological consequences of DFADs on tuna ecology and the marine environment is required. The mechanics of how and why tuna connect with DFADs are currently unknown. Several hypotheses have been proposed to explain tuna aggregation to DFADs [44], [66], [67], including the “meeting point” hypothesis, which proposes that DFADs facilitates the encounter between individuals or schools, resulting in larger aggregations that may benefit survival rates [44], and the “indicator log” hypothesis, which proposes that tunas may use drifting objects as indicators of areas with abundant plankton and food to ensure the survival of their eggs, larvae, and juveniles [68]. Some author have proposed that man-made DFADs might form an “ecological trap”, causing tuna to remain linked with DFADs even as they migrate into locations that could severely effect tuna behaviour and biology [69], [70]. However, there is currently insufficient data to validate or refute this idea (see Dagorn, Holland, Restrepo *et al.* [71] and related references).

Today, DFADs utilised by tropical tuna purse-seine fisheries is frequently deployed with satellite-linked instrumented buoys fitted with one or more echo-sounders, which offer precise DFAD placement and estimates of aggregated tuna biomass [46], [54]. The data acquired by these buoys is useful to fishermen, but it has also piqued the interest of scientists, who have recognised their potential to give insights on tuna migration and behaviour on a worldwide scale [57], [58], [60], [62]. Recent research works have employed echo-sounder data from these buoys to remotely map tuna distribution or explore trends in tuna aggregation around DFADs sites [57], [62], [72]. According to some authors [73], DFAD data may be utilised to evaluate the ecological trap hypothesis.

1.3. Weather routing

Ship weather routing is defined as [74] “determining the optimum route of a ship that utilizes the optimum engine speed and power for the ocean voyage based on weather forecasts, sea conditions and the individual characteristics of the ship”. The term “optimum” has multiple meanings, such as maximizing ship safety and crew comfort, reducing fuel consumption, Green House Gases (GHG) emissions and duration of the voyage, or any combination of these goals. Ship weather routing is an interdisciplinary problem and has attracted the attention of ocean and naval engineers, computer scientists, data scientists, maritime economists and transportation and logistics engineers [75]. In fact, in recent years both academia and industry had shown an increasing interest on this topic. Two main factors are responsible for the recent explosion of optimization algorithms for marine routes with weather and ocean information:

- The recent availability of very fine grained information on waves and oceanic currents, at a resolution of 0.08° for ocean data (and even finer scales at specific regions), based on Ocean General Circulation Model (OGCM), publicly available at data repositories provided by Copernicus Marine Environment Monitoring Service (CMEMS), Hybrid Coordinate Ocean Model (HyCOM) or NOAA. These data are updated daily and include a reliable forecast into the future.

- The increasing amount of data on ship consumption allows finer models for a specific vessel under different cruising speeds, sea and weather conditions.

Aside from these technical improvements, a main driving force is the regulatory pressure from the International Maritime Organization (IMO) via the recent adoption of the Energy Efficiency Design Index (EEDI) to **reduce emissions** related to sea transportation by 40% in 2030 [76]. In order to compute their EEDI, from 2018 all IMO vessels are obliged to record their fuel consumption [77]. Despite being the most efficient way to transport freight [78], maritime transport is still responsible for almost 3% of global Green House Gases (GHG) emissions, and this number is bound to increase if no actions are taken [79], [80]. This pressure is triggering much research into alternative fuels for the shipping industry [81], and the current regulation entails that some of the older tankers in the fleet will no longer be allowed to operate. Weather routing is proposed as another mean to reduce emissions, by increasing the performance of the ship [82].

The **economic factor** is another reason why weather routing research has spurred. Nowadays, over 78% of international good exchanges are done through maritime transport [83]. Around 60% of the total operating costs of sea transportation is the cost of fuel, for a total bill of 120 billion EUR per year for the whole merchant fleet [84]. The estimated savings of 3-5% thanks to better routing [75] would have a huge economic impact. In addition, ship weather routing also addresses **safety issues**. Around 48% of the accidents at sea involve human casualties or loss of cargo are due to sailing under adverse weather conditions [85].

One last factor worth mentioning is the emergence of **slow steaming** during 2009 crisis [86]. This concept revolves around the non-linear relation between speed and consumption: reducing a vessel's speed by 10% decreases emissions and fuel consumption by at least 10–15% [87]. With slower ship speeds, ocean currents, wind and waves become even more relevant making weather routing systems a necessary tool for marine transportation. In particular, slow steaming can take a big advantage of wind [88] by installing Wind Assisted Ship Propulsion System (WASP) - such as sails, wing-sails, e-sails or Flettner rotors - on ships. A reliable weather routing algorithm is necessary for correctly assessing the impact of retrofitting working cargo ships to install WASP. The viability analysis of WASP is still carried out considering a single fixed route, namely the actual route that the captain chose when sailing on fuel power alone [89], even though anyone who has sailed on wind energy alone knows that the optimal route for a sailboat depends strongly on the wind's direction. The implication of this observation is that if WASP are combined with weather routing, there is a lot of room for further savings.

Even with the rising interest on ship weather routing and all the benefits involved, developing a state of the art weather routing system remains a challenging task. There are several factors to take into account to ensure accurate savings on fuel and GHG emissions [90]. First, the models for fluid-structure interaction and fuel consumption have high complexity, and depend heavily on the type of ship. Second, there is the multi-scale and time varying weather and ocean patterns. Last, the computational requirements to handle effectively all the necessary data are huge, and the complexity of the mathematical optimization problem makes it even more challenging.

1.4. Objectives

The introductions of sustainable fishing and weather routing done in Sections 1.2-1.3, have led us to enumerate the following three major challenges, object of study in this thesis.

1.4.1. Objective 1. Tuna estimation using several sources of data

In Section 1.2.1 we commented on how fisheries often use DFADs to attract tuna schools, and echo-sounder buoys attached to those DFADs can estimate the amount of tuna. However, we also mentioned that these estimations often do not correlate well with the real amount of tuna caught by the fishing vessels. Some studies have tried to improve these estimations by using the echo-sounder data, while others used oceanography, but we found no study that combined both sources. Thus, the first challenge in this thesis is to establish a well defined pipeline for estimating tuna biomass under DFAD echo-sounder buoys, combining catch data, oceanography and echo-sounder buoy data.

Addressing this challenge first requires the application of DS methods to study and preprocess the data from all sources mentioned previously. It also involves building and training different ML models and defining relevant metrics to compare them against each other, in order to find the algorithm that best estimates the tuna biomass under a DFAD using the available information.

1.4.2. Objective 2. Study tuna dynamics using ML models

Once **Objective 1** is achieved, we will have a ML pipeline specialized in measuring the tonnage of tuna associated to any DFAD. We could then apply that pipeline to perform biomass estimations in large scale and for long periods of time across all oceans, and characterize the temporal patterns of tuna associations to DFADs. These analysis would allow us to test the hypothesis mentioned in Section 1.2.2: how long does tuna associate to DFADs and whether this aggregations remain over time or rather disaggregate.

1.4.3. Objective 3. Test weather routing algorithms

In Section 1.3 we explained the concept of weather routing, why it is relevant and how it can be addressed through a DS approach, thanks to the huge amounts of oceanographic data available today. The last challenge in this thesis is to test different weather routing algorithms, that must be able to provide optimum routes under varying weather conditions.

Meeting this challenge is a complex task and must be divided into smaller goals. First, a DS study should be conducted to collect and process the weather data, and implement a consumption model. Then, we must find one or more optimization algorithms able to compute routes that minimize travel time or fuel consumption, using weather data. There exists several optimization methods, that will be explained on Chapter 4. Finally, we should test these optimization algorithms in both synthetic benchmarks and real-life scenarios, comparing its results against some reference routes.

1.5. Thesis structure

Chapters 2 and 3 elaborate into the concept of sustainable fishing applied to tuna, addressing Objectives 1 and 2 respectively. These chapters describe a project carried in collaboration with Komorebi AI and Satlink. As no official name was given to the project, both chapters are named after the ML pipeline developed to study tuna dynamics: TUN-AI . Chapter 4 addresses Objective 3 and presents the second study. It is an application of weather routing to reduce emissions and save fuel during cargo boat operations. Finally, Chapter 5 gathers concluding remarks and outlines open research problems for both fields. Figure 1.4 shows how this thesis is structured.

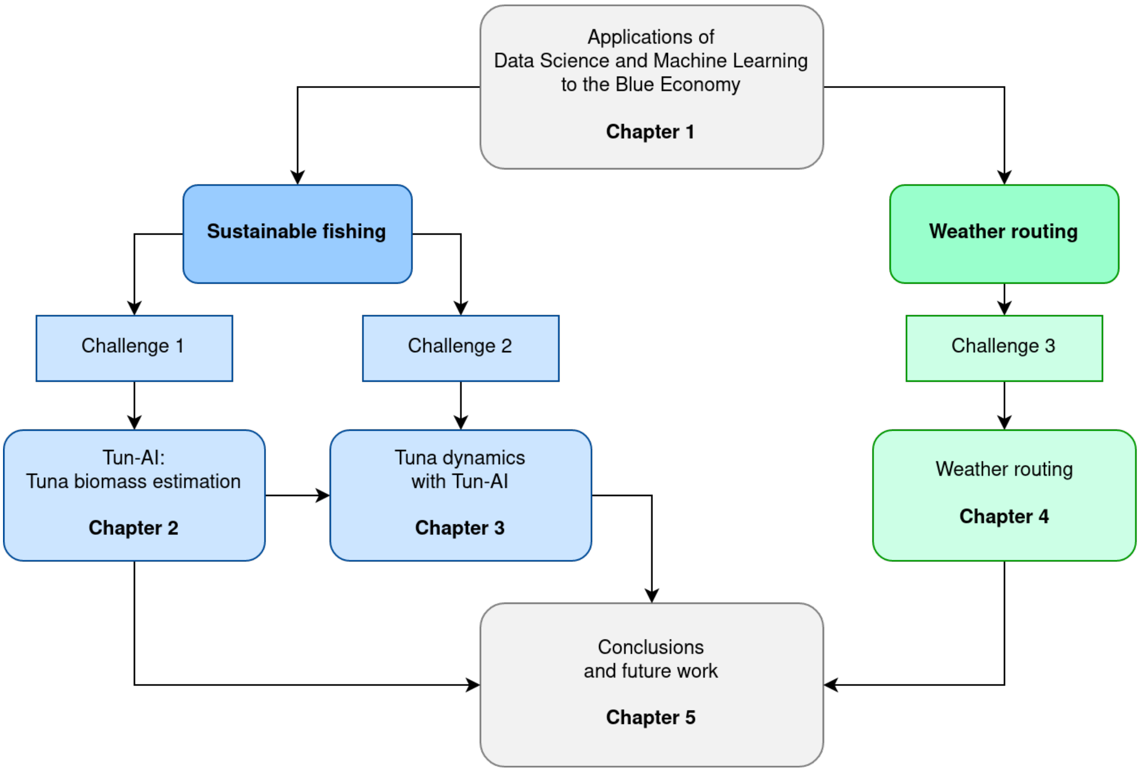


Figure 1.4: The structure of this thesis.

2. TUN-AI: TUNA BIOMASS ESTIMATION

2.1. Introduction

Tuna purse-seine fisheries often use Drifting Fish Aggregating Devices (DFADs) equipped with echo-sounder buoys to estimate the biomass of tuna aggregated around the devices. While this information can provide valuable insights into tuna behavior and abundance, it can be challenging to analyse and utilize. The study described in this chapter aims to develop a pipeline, called TUN-AI , that utilizes DFAD logbook data, echo-sounder buoy data and oceanographic data, to evaluate various Machine Learning (ML) models for processing these data sources and estimating tuna biomass in metric tons (t). The models include binary classification, ternary classification, and regression, and the goal is to establish a method for accurately predicting tuna biomass at different levels of complexity.

The structure of this chapter is as follows. First, the state of the art in tuna biomass estimation is introduced in section 2.2. Then, section 2.3 describes the data sources available for this study and the design of TUN-AI pipeline, including how the information was processed and the ML models used to estimate tuna biomass and presence. Results are shown in section 2.4 and compared with the state of the art in section 2.5.

2.2. State of the art

There are not many studies that estimate tuna presence or biomass, and even less than do so by applying ML. One reason is that studying tuna populations often requires specialized equipment and expertise, which can be expensive and difficult to obtain. In addition to this challenge, each study often uses different metrics to measure tuna populations. This lack of consistency makes it difficult to compare the results of different studies and to draw meaningful conclusions about the overall state of the art.

This section reviews the few works found on tuna estimation, and groups them into two categories: studies done to predict tuna presence, that apply classification models, and works that estimate the tons of tuna by applying regression models.

2.2.1. Classification models for tuna presence

Studies that apply classification models focus on predicting whether tuna is present or absent, with no estimation on the school size. Tuna is considered present under the DFAD when there is a relevant amount of tuna biomass (typically, more than 1t), and is considered absent otherwise.

Based on tuna presence and absence data observed by scientists, Uranga, Arrizabalaga, Boyra *et al.* [91] created a reference dataset of sonar images with known categories of “tuna” or “no tuna”. They used these images to validate their classification and analysis procedure. The sonar’s searching range varies depending on sea conditions and skipper preferences, but in general, a range of 100-300 metres was used when looking for tuna. The images were taken during two scientific tuna surveys in 2009 and 2011. The authors extracted 22 501 blobs from the 1397 presence and 1398 absence images collected to create the training dataset: 1497 were positive examples (presence) and 21 004 were negative examples (absence). They used five different classification models on the dataset: Random Forest (RF) [92], Support Vector Machine (SVM) [93], Multi-Layered Perceptron (MLP) [94], Iterative Dichotomiser 3 (ID3) [95] and Instance Based learner with fixed neighbourhood (IBK) [96]. The classification models’ results were validated using 10-fold cross-validation for 30 different runs, and the results are shown in Table 2.1.

Baidai, Dagorn, Amande *et al.* [57] uses RF classification to convert acoustic backscatter from echo-sounder buoys into tuna presence metrics. Training datasets were created by combining acoustic data with logbook and observer data from the Atlantic and Indian Oceans that reported DFAD activities (tuna catches, new deployments, and DFAD visits) from 2013 to 2018. Under the DFAD, tuna was considered present if at least 1t was caught nearby. Table 2.1 displays their results.

Up to this point, we have only reviewed works that use the echo-sounder information derived from the DFAD to perform the estimations. However, oceanography data is another motivator for some studies attempting to model tuna behaviour. Druon, Chassot, Murua *et al.* [64], for example, used Chlorophyll-a fronts as a proxy for food availability and described tuna environmental preferences using a combination of other oceanography features. Using this data, the authors developed an Ecological Niche model that predicts favourable habitat zones for skipjack tuna (*Katsuwonus pelamis*). The researchers compared these predictions to presence data collected by the European purse seine fleet around DFADs over a 15-year period. They discovered that fishing sets occurred 34% (45%) of the time inside the estimated favourable habitats in the Atlantic (Indian) Ocean. Their scores are shown in Table 2.1.

Another study worth mentioning is Mannocci, Baidai, Forget *et al.* [97], who also explored the possibility of applying ML on echo-sounder buoys attached to DFADs. In this case, rather than predicting tuna presence, the authors trained RF algorithms to differentiate between high and low by-catch occurrence (i.e. incidentally capturing species that are not tuna). Their models were based on matched echo-sounder and onboard observer data for the same DFADs. This study was conducted in the Atlantic and the Indian Ocean, with sample sizes of 838 and 2144 respectively. Algorithms showed a better performance in the Atlantic Ocean (see Table 2.1) and were best at detecting the “high by-catch” occurrence class. This study proves that it is possible to use the same sources of data (in this case, acoustic records for echo-sounder buoys) and ML architectures to

Ocean	Model	N	Accuracy	Recall	Specificity	F_1-score
Atlantic	EN [64]	3159	0.34	-	-	-
	RF [91]	22 501	-	0.79	0.99	-
	SVM [91]	22 501	-	0.79	0.99	-
	MLP [91]	22 501	-	0.77	0.99	-
	ID3 [91]	22 501	-	0.73	0.99	-
	IBK [91]	22 501	-	0.75	0.98	-
	RF [57]	1856	0.76	0.83	0.67	0.78
	RF [97] [a]	838	0.66	0.74	0.59	0.67
Indian	EN [64]	26 143	0.47	-	-	-
	RF [57]	13 671	0.85	0.81	0.90	0.84
	RF [97] [a]	2144	0.58	0.63	0.53	0.60

[a] Predicts by-catch presence during fishing sets for tuna.

Table 2.1: Several approaches to predict tuna presence seen in the literature. Their predictions were compared against real fishing set data. For each model, it is shown the available scores and the number of samples (N) it was compared against. Each sample is a single binary value: whether a substantial amount of tuna was there or not. Metrics are explained in Appendix 2.A.

predict different targets (for instance, by-catch presence instead of tuna abundance).

2.2.2. Regression models for tuna biomass

Some works apply regression models to estimate the amount of tuna under the DFAD. These predictions are often done in tons of tuna, and compared against real fishing set data from fishing vessels.

Lopez, Moreno, Boyra *et al.* [60] developed a behaviour based approach to provide relative biomass estimates of fish aggregations at DFADs in the Atlantic Ocean. Their model uses data from Satlink, one of the most common brands of echo-sounder buoys. For this study, the acoustic samples were collected at sunrise, because according to the belief of fishermen it is the time when fish are more concentrated under the DFAD. Acoustic information, divided by layer depths, was used to derive rough biomass estimations following the target strength of tuna species and their studied vertical distribution (i.e. what depths are they expected to habit). The rough biomass estimations were later corrected through different regression models, namely Generalized Linear Model (GLM) [98], polynomial of order 2 and 3 (POL2, POL3 respectively) and Generalized Additive Model (GAM) [98]. These regression models were fitted by using 21 real fishing sets, and the final estimations were also compared against those catches. All their scores are shown in Table 2.2. For comparison, the authors also provided the biomass estimates from the buoys' manufacturer (Satlink), whose models were calibrated for the target strength of skipjack tuna (*Katsuwonus pelamis*).

Following Lopez, Moreno, Boyra *et al.* [60] proposed models, Orue, Lopez, Moreno *et al.* [99] replicated a similar study at the Indian Ocean. Their models were once again build based on existing knowledge of the vertical distribution of non-tuna and tuna at DFADs and mixed species target strengths and weights. This study used information from 287 fishing sets and their corresponding acoustic samples from echo-sounder buoys prior to the fishing set. Results showed that manufacturer's biomass estimates generally improve. However, the authors observed that the results obtained by Lopez, Moreno, Boyra *et al.* [60] in the Atlantic Ocean were significantly better than theirs, albeit the number of samples used for the analysis was much lower in that case (21 samples against their 287). Refer to Table 2.2 for a comparison of scores between both studies. The authors also commented that the improvement of the biomass estimates was not as large as they expected, indicating that the large spatial temporal variability in the Indian Ocean is not easily considered with a linear model.

Ocean	**Model**	**N**	**R^2**	**MAE**
Atlantic	Manufacturer [60]	21	0.25	21.29
	GLM [60]	21	0.73	-
	POL2 [60]	21	0.81	-
	POL3 [60]	21	0.83	10.25
	GAM [60]	21	0.82	-
Indian	Manufacturer [99]	287	0.02	-
	GLM [99]	287	0.02	-
	POL2 [99]	287	0.02	-
	POL3 [99]	287	0.03	-
	GAM [99]	287	0.03	-
Pacific	SEAPODYM [100]	300	0.85	-

Table 2.2: Several approaches to estimate tuna biomass seen in the literature. Their predictions were compared against real fishing set data. For each model, it is shown the available scores and the number of samples (N) it was compared against. Each sample is a single value, usually the tons of tuna reported from a fishing set. Metrics are explained in Appendix 2.B.

Studies on regression models mentioned until this point only use the acoustic data from echo-sounder buoys. However, tuna dynamics is known to be affected by oceanography conditions (such as temperature and dissolved oxygen) and there exists works that model tuna behaviour from oceanography data. Lehodey, Senina and Murtugudde [100] proposed a spatial ecosystem and population dynamics model (SEAPODYM), based on advection - diffusion - reaction equations. The model was tuned using empirical data from commercial fisheries at Western and Central Pacific Ocean, and applied to two tuna species: skipjack (Katsuwonus pelamis) and bigeye (Thunnus obesus). By taking into account the climate variability, SEAPODYM predicts both temporal and spatial distribution of tuna populations over 1° grid-boxes at monthly resolution. Validation of the model against observed catch (from the same commercial fisheries, accumulated by month) showed an R^2 of 0.85 [101], see Table 2.2. However, it must be noted that setting the value of each parameter in SEAPODYM was performed mostly by *ad hoc* manual "tuning", using independent models estimates and by application of parameter values gleaned from the scientific literature. Thus, SEAPODYM application is limited, as it needs to be manually tuned in order to use it at other ocean regions or for different tuna species.

2.2.3. Contribution

This chapter presents a new approach for estimating tuna biomass using ML models that incorporate oceanographic data as predictor variables, in addition to catch data and echo-sounder buoy information. This approach, referred to as Tun-AI , aims to accurately estimate tuna biomass in metric tons (t) under DFADs equipped with echo-sounder buoys at any given time. The study evaluates the performance of various models, including binary classification models that distinguish between tuna biomass less than 10t and greater than or equal to 10t; three-level classification models that differentiate between tuna aggregations less than 10t, between 10t and 30t, and over 30t; and regression models that estimate the exact tuna biomass in tons. We also examine the influence of different data sources and methods for processing echo-sounder buoy data to identify the most accurate methodology. This research represents an improvement on previous studies that have only compared biomass estimates from buoys to catch data [57], [60], [97], combined oceanographic variables and catch data without using echo-sounder buoy information [64], [100], or considered the effects of oceanographic conditions on buoy biomass estimates without directly comparing them to catch data [59], [63].

2.3. Material and methods

2.3.1. Database description

Our study draws from three sources of information: FAD logbook data, echo-sounder buoy data, and oceanography data.

FAD logbook data

The first database contains data on the activities of the Spanish tropical tuna purse seine fleet on DFADs in the Atlantic, Indian, and Pacific Oceans. These figures were provided by the shipowners' association, Asociación de Grandes Atuneros Congeladores (AGAC), and cover a time period from 11th April 2017 to 1st January 2021. In total, 120 707 events have been recorded in the database, with 35 813 occurring in the Atlantic, 55 819 in the Indian Ocean, and 29 075 in the Pacific. Each database entry includes information about the type of interaction with the DFAD, the identification number and model of the echo-sounder buoy attached to the DFAD, the timestamp and location of the activity, and other pertinent information. See Ramos, Báez, Grande *et al.* [102] for more information on the various types of interactions.

We used the identification number for the buoy attached to the DFAD to match the human interactions recorded in our database to the echo-sounder acoustic measurements (as described in Section 2.3.1). We only considered interactions labelled "Set" and "Deployment" in our analysis and compared them to data from the echo-sounder buoy. The "Set" data included catch information for skipjack, yellowfin, and bigeye tuna, which we used to represent the actual tuna biomass at the DFAD. This assumption is based on the belief that fishing vessels captures the entire tuna aggregation at the DFAD during a set and record the total catch accurately. While this assumption is strong, it is necessary when working with large-scale data sets like ours. We did not incorporate by-catch data in our analysis, despite the fact that it is recorded in the logbook.

It should be noted that purse seine vessels only deploy their nets when they receive information about the presence of large tuna biomass, resulting in low catch rates (less than 10t) being uncommon (representing less than 8% of all interactions). This indicates that the dataset from "Set" events may not precisely reflect the genuine data distribution, potentially leading to models that overstate true tuna biomass and have limited real application. Table 2.1 depicts the distribution of catch quantities. To remedy this issue, we included "Deployment" interactions, which represented new DFADs that were not previously present in the water [102]. We assumed that no tuna (0t) were present under the buoy throughout these interactions.

Misreported positions (either latitude or longitude), inaccurate echo-sounder buoy IDs, dates and times, and incorrect interaction types can all occur during the manual transcription procedure for the FAD logbook. To lessen the impact of these inaccuracies, we used the buoy ID and timestamp to cross-reference the information in the FAD logbook to the echo-sounder data to identify any differences (see Section 2.3.2).

Echo-sounder buoy data

The data for the echo-sounder buoys came from 16 419 Satlink buoys that had documented interactions in the FAD logbook. This database includes approximately 70 million records for DFAD buoys in the Atlantic, Indian, and Pacific Oceans. From 2018 to 2020, each record is associated with a specific buoy ID and timestamp and includes biomass estimates based on echo-sounder measurements as well as GPS coordinates for the buoy's last known location at the time of measurement.

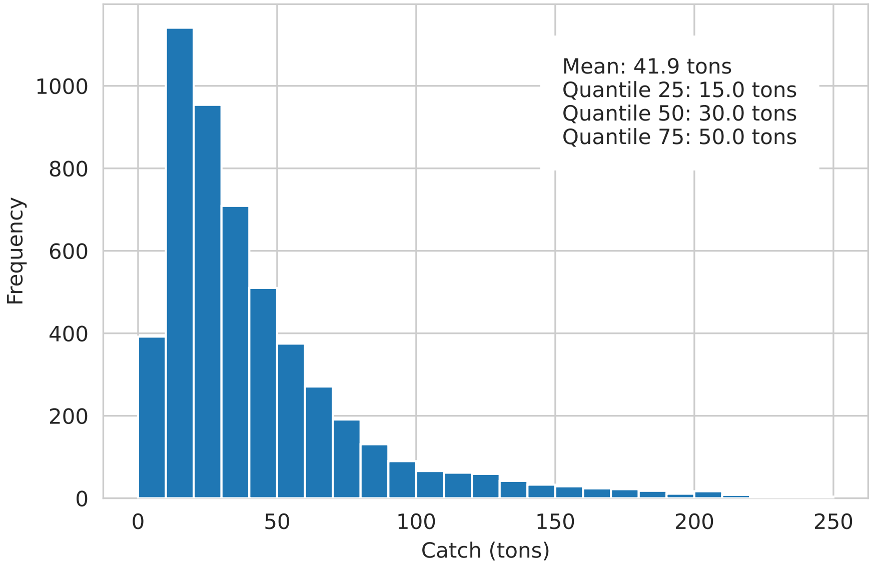


Figure 2.1: Tonnage distribution of tuna captured from a total of 5202 sets.

The echo-sounder observation range for all buoys in the study is 3 to 115 metres in depth, separated into ten layers or depth bins with a resolution of 11.2 metres (see Figure 2.2). Estimated biomass in metric tonnes is generated from acoustic samples taken at regular intervals throughout the day and converted into estimated tonnage based on skipjack tuna target strength. Appendix 2.C and Boyra, Moreno, Sobradillo *et al.* [103] contain additional information about the buoy's process and the target strength value, respectively.

All of the buoy models employed in the study (ISL+, SLX+, and ISD+) use the same technique for translating acoustic response to estimated tonnage, however there are minor changes between them, particularly in terms of sampling rate (i.e., the frequency of echo-sounder measurements; see Table 2.3). Each buoy has an echo-sounder that takes multiple measurements every hour, but in order to reduce the amount of data sent via satellite, only the measurement that corresponds to the highest estimated tonnage per hour is transmitted by the buoy and stored in central databases. If the total estimated tonnage for all measurements collected during an hour is less than 1t, no measurement is transmitted and the reading is regarded zero. Similarly, due to echo-sounder signal saturation, the total estimated tonnage for a single depth bin is limited to 63t. As a result, the final temporal resolution of the echo-sounder records in the current dataset is 1 hour.

Oceanography data

Oceanographic data from the European Union's CMEMS [104] was utilized in this study (products GLOBAL ANALYSIS FORECAST PHY-001-024 and GLOBAL-ANALYSIS-FORECAST-BIO-001-028, both with a resolution of $1/12°$ and $1/4°$, respectively). The following variables were extracted for each position record in the echo-sounder buoy data

Buoy model	ES model	Frequency	Beam angle	Sampling rate
ISL+	ES12	190.5kHz	20°	Every 15min
SLX+	ES16	200kHz	23°	Sunrise to sunset: every 5min Sunset to sunrise: every 60 min
ISD+	ES16x2	200kHz and 38kHz [a]	23° and 33°	Sunrise to sunset: every 5min Sunset to sunrise: every 60min

[a] Data can be compared across all buoy models because biomass estimations are based on the acoustic response captured by the 200kHz echo-sounder.

Table 2.3: Models and characteristics of buoys. ES is abbreviation of "Echo-sounder".

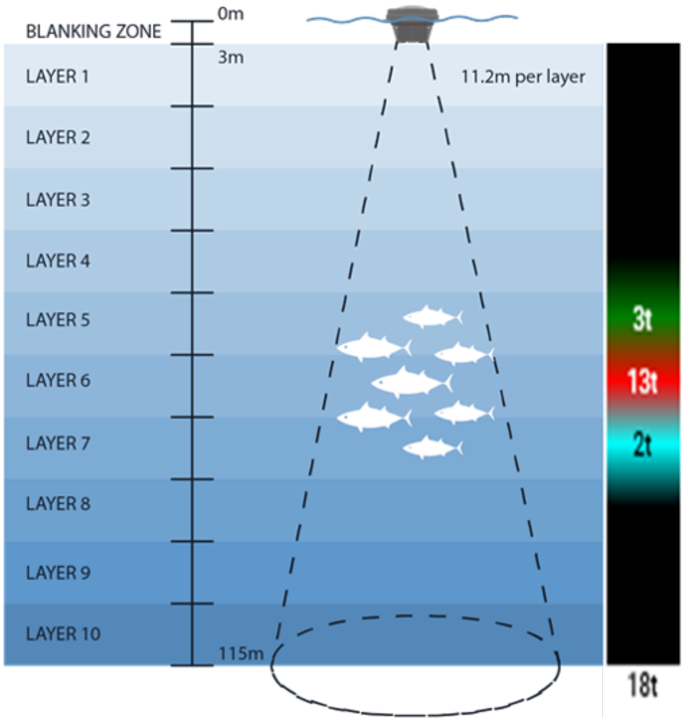


Figure 2.2: Left: Depth layer configuration and Satlink echo-sounder buoy setup. Right: An example of the biomass estimates (in metric tonnes) and echo-gram display that buoy users can access. Using the manufacturer's algorithms, raw acoustic backscatter is converted into biomass estimates based on the target strength of skipjack tuna (*Katsuwonus pelamis*).

(see Section 2.3.1): temperature (°C), chlorophyll-a concentration (mg/m^3), dissolved oxygen concentration (mmol/m^3), salinity (psu), thermocline depth (calculated as the depth where the water temperature is 2°C lower than the surface temperature, in m), current velocity (m/s), and Sea Surface Height anomaly (SSHa) (deviation of the sea surface height from the long-term mean, in m). All variables, except for thermocline and SSHa, were obtained at the surface level (depth = 0.494m).

It should be noted that the aforementioned variables are derived from oceanographic models, which provide approximate values on a fixed grid rather than actual observations at the exact position of the buoy. However, because the accuracy of these models has improved in recent years, we believe that these values are sufficiently representative of the oceanographic conditions for the purposes of the current study (see, for example, Lellouche, Greiner, Le Galloudec *et al.* [105]).

2.3.2. Data preprocessing

Data merging

Using the buoy ID and timestamp, the sets and deployments registered in the FAD logbook data were cross-referenced with each specific buoy's biomass estimates. After that, oceanographic data was collected for each position recorded in the echo-sounder buoy database. Oceanographic data is available on a grid with 0.08° or 0.25° resolution, so we incorporated data from the point on the grid closest to the buoy's position. We assume that oceanographic variables change on a larger spatial scale than grid spacing and buoy hourly movement, so this approximation has no significant errors.

Echo-sounder window

Tuna schools exhibit well-known circadian behavior near the DFADs, arriving at or near sunrise and departing around sunset, staying close to the DFAD for several days in a row [106], [107]. We include a large enough window of echo-sounder measurements with hourly frequency as an input to the model to capture these patterns. We investigated how the inclusion of time windows of varying lengths (24, 48, or 72 hours) affected the model's ability to correctly estimate daily tuna biomass. The echo-sounder window length that produced the best results was used in all subsequent analyses.

For set events, the selected window ends at sunset the day before the event and begins 24, 48, or 72 hours before that (see Figure 2.3 for an example). Starting at sunset aligns all observations with solar time regardless of time zone, and ensures that all echo-sounder measurements in the window are taken before the set event, regardless of when it occurred during the day.

For deployment events, because the buoy is not yet in the water, we cannot take echo-sounder measurements prior to the event. As a result, the echo-sounder window is chosen after the deployment, again in accordance with solar time. As stated in Section 2.3.1, we assume that no tuna is aggregated during the first 1-5 days after deployment, based on findings from Orue, Lopez, Moreno *et al.* [62].

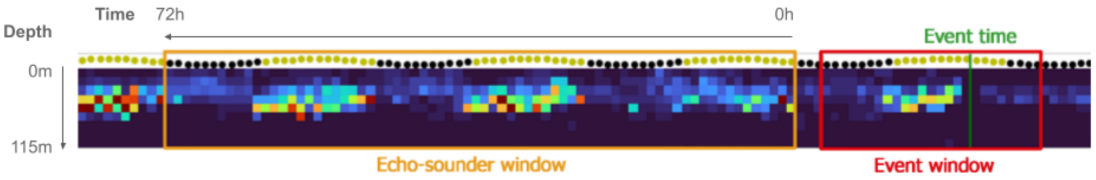


Figure 2.3: An example of a 72-hour “echo-sounder window” (yellow box) in relation to the recorded set time in the FAD logbook (green line). The day of the event (24 hours) is referred to as the “event window” (red box). It is worth noting that the yellow and red boxes do not overlap. The sun’s inclination throughout the day is depicted above the graph, with day hours represented by yellow circles and night hours represented by black circles. The echo-sounder buoy biomass estimates for each hour are represented by columns of coloured squares, while rows are depth bins. This figure clearly shows the circadian patterns in tuna activity under the DFAD.

Data cleaning

As previously stated, the data used in this study may contain errors, particularly in the case of FAD logbook data, which is recorded manually. To reduce the possibility of errors, the following conditions had to be met in order for an event (set or deployment) to be included in the final dataset:

- The buoy ID in the FAD logbook data must match the buoy ID in the acoustic database, ensuring that echo-sounder data are available for the DFAD on which the event occurred. This avoids issues where the buoy ID is misreported in the FAD logbook, unless the faulty ID happens to match the one from another buoy.
- The windows described in Section 2.3.2 from one event cannot overlap with the windows from another event. For example, we exclude from our analysis sets events that occurred within a few hours of each other. This requirement is in place to ensure that there is no human intervention on the DFAD during the window of echo-sounder measurements used for estimation.
- Events with invalid positions (for example, buoys on land) were removed from the dataset using the same criteria as Escalle, Heuvel, Clarke *et al.* [61].
- Events or measurements recorded at locations with less than 200m of water depth were eliminated. This avoids incorporating echo-sounder measurements that could be impacted by the seabed.
- We estimated buoy speed for each position using the last known location of the buoys, and deleted occurrences and measurements where the buoy speed was greater than 3 knots, because surface currents in the tropical oceans rarely reach this speed [62]. This avoids including measurements collected on a ship that are not indicative of a DFAD.

The final dataset, after merging and filtering, contains over 12 000 events. These interactions occurred on 10 063 buoys, for which over 665 000 echo-sounder records were collected. Table 2.4 clearly shows that events were more or less uniformly distributed by ocean, allowing for subsequent analyses stratified by ocean basin. However, for the buoy model, where the majority of interactions happened on ISL+ buoys, similar experiments could not be undertaken with the available data.

	Ocean basin			Buoy model			
	Atlantic	Indian	Pacific	ISL+	SLX+	ISD+	Total
Set	1500	2727	974	4877	192	132	5201
Deployment	1369	2199	3426	6443	297	254	6994
Total	2869	4926	4400	11 320	489	386	12 195

Table 2.4: Number of Set and Deployment events remaining after integrating echo-sounder and FAD logbook data, per ocean, each buoy model, and in total.

2.3.3. Model selection

We tested multiple models with varying feature sets to determine the relative contribution of different features to model accuracy as well as the overall performance of various modelling strategies.

Baseline model

As a baseline, we created a model utilising solely the biomass estimations from the echo-sounder window (see Section 2.3.2). However, because the model's output (i.e., the total biomass estimation) would be a single number, a series of aggregation criteria had to be applied to the $72 \times 10 = 720$ echo-sounder window matrix. These values can be aggregated in two ways (see Figure 2.4): by rows (layers) or by columns (hours), with various aggregation criteria. We decided to test all conceivable combinations of aggregation rules (mean, maximum, and sum) and aggregation directions (by hour and depth bin or layer), selecting the one with the lowest Mean absolute error (MAE), that is, the one that can best estimate the tonnes of tuna taken under the DFAD.

It is relevant to notice that some combinations must result in the same MAE, and only one of them is reported in these circumstances. For example, if the aggregation function by layers and hours is the same, the sequence in which these aggregations are conducted is meaningless. The same is true for aggregations that simply contain the total and the mean, because they are both linear functions.

Table 2.5 contains the results for all eligible combinations. As an example of how to read this table, consider the best performing aggregation: layer, max, mean. To acquire

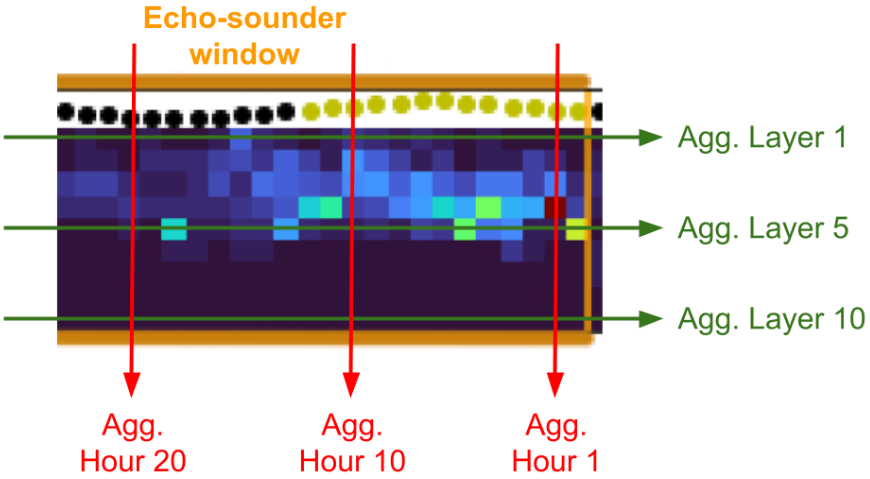


Figure 2.4: An illustration of how biomass measurements are aggregated.

First aggregation direction	First aggregation function	Second aggregation function	MAE
`layer`	`max`	`mean`	13.86
`layer`	`mean`	`sum`	13.98
`hour`	`mean`	`max`	15.64
`hour`	`max`	`mean`	15.69
`layer`	`max`	`max`	16.47
`layer`	`mean`	`max`	16.64
`layer`	`mean`	`mean`	17.59
`hour`	`sum`	`max`	28.42
`layer`	`sum`	`mean`	37.67
`layer`	`max`	`sum`	59.91
`layer`	`sum`	`max`	143.78
`hour`	`max`	`sum`	243.02
`layer`	`sum`	`sum`	483.00

Table 2.5: Different data aggregations for the regression baseline model.

the final forecasts for this aggregation, first take the maximum value of the 72 hours for each layer. This yields ten numbers, one for each layer. Then we combine these ten values by computing the average, yielding a single number. The baseline for the classification models is immediately computed from the preceding estimate, just by comparing the output to the defined thresholds.

Feature engineering

We considered the variables in Table 2.6 as features to be included in each model based on the merged dataset (see Section 2.3.2). Depending on the size of the echo-sounder window, the original biomass measurements form a $10 \times \{24, 48, 72\}$ matrix. These values were not directly fed into the models, but were aggregated using the rules shown in Figure 2.5:

- Using the maximum by row (layer). This yields a vector of size 10.
- Using the maximum by column (hour). Depending on the size of the echo-sounder window, this produces a vector of size 24, 48, or 72.
- By layer and by hour, computing the maximum of the 24, 48, or 72 hours, followed by the mean by layer. This yields a single value. The maximum and then the mean is chosen because that combination was the best performing baseline model (see Section 2.3.3).

These vectors (along with the baseline model) were then used directly as features in the various models. Depending on the size of the echo-sounder window, the total number of echo-sounder variables was then $\{24, 48, 72\} + 10 + 1$. The previous feature vector generation procedure is applied to both set and deployment events.

Task description

Models were trained to perform four different tasks, which are described below in order of increasing complexity:

1. A binary classification problem in which the target variable y (tuna biomass) might be either $y <$ 10t or $y \geq$ 10t.
2. A ternary classification task in which the target variable y (tuna biomass) might have values of $y <$ 10t, 10t $\leq y <$ 30t or $y \geq$ 30t.
3. A threshold regression task in which we directly estimated tuna biomass y in metric tonnes up to a threshold of 100t. Estimates equal to or higher than 100t were clipped.
4. A regression task in which we directly estimated the tuna biomass y in metric tonnes.

The thresholds used to determine the categories were chosen using a variety of criteria. In all classification tasks, the lower threshold was based on best-practice standards for reducing shark by-catch, which include avoiding sets on tuna schools weighing less than

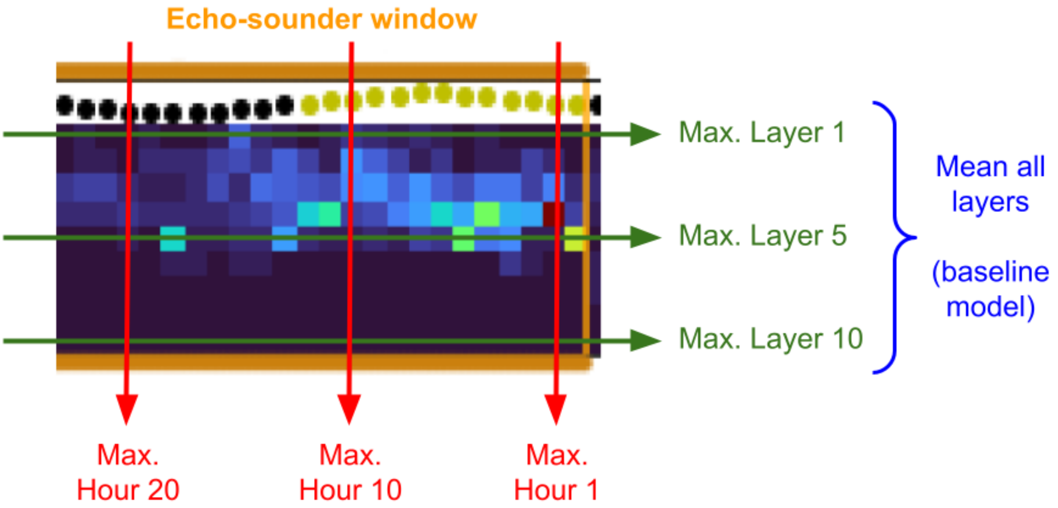


Figure 2.5: A visual representation of how the biomass readings were aggregated. Columns of coloured squares represent the echo-sounder buoy biomass estimates for each hour, while rows represent depth bins or layers. The value of the estimated biomass is represented by the square colour. First, the maximum value for each layer is calculated, yielding a vector of size 10. (in green). Second, depending on the size of the echo-sounder window, the maximum value for each hour is calculated, resulting in a vector of size{24, 48, 72}. (in red). Finally, the baseline model (in blue) was fed into the ML models as an input.

10t [71]. The second class in the ternary classification task was further split by the dataset's median catch (30t). In the threshold regression task, we chose 100t because sets above that were quite infrequent (315 occurrences, 8.1%). Table 2.1 shows the entire distribution of the tonnes of tuna caught in the specified events.

Machine Learning models

We partition the dataset training (75%, 9152 events, 3893 sets and 5259 deployments) and test (25%, 3051 events, 1309 sets and 1742 deployments) as is customary in supervised ML, while keeping the whole class distribution. Because the number of observations was comparable across oceans (see Table 2.4), we did not stratify these divisions by ocean. In the classification and regression tasks, we compared the performance of a baseline rule-based model (see Section 2.3.3) versus five distinct ML models:

- Logistic Regression (LR) classifier [108]: a linear model for the classification task.
- Elastic Net (ENet) regressor [109]: for the regression task, with three regularization techniques, namely $L1$ penalization, $L2$ penalization and elastic net.
- Random Forest (RF) algorithm [92].
- Gradient Boosting (GB) algorithm [110].

	Echo	Echo + Ocean	All
`Biomass measurements`	✓	✓	✓
`Number of zero-readings`	✓	✓	✓
`Buoy model`	✓	✓	✓
`Chlorophyll-a`		✓	✓
`Dissolved oxygen`		✓	✓
`Salinity`		✓	✓
`Thermocline depth`		✓	✓
`Temperature`		✓	✓
`Current velocity`		✓	✓
`SSHa`		✓	✓
`Day and month`			✓
`Year`			✓
`Latitude`			✓
`Longitude`			✓
`Ocean basin`			✓
`Sunrise hour`			✓
`Sunset hour`			✓

Table 2.6: Grouped features used for the models. “Echo” contained only data from the echo-sounder buoy database relating to echo-sounder measurements (in blue). “Echo + Ocean” included oceanographic data for each record in the echo-sounder buoy database’s position and date (in green). “All” contained additional data derived from each record in the echo-sounder buoy database, such as position and time (in red).

- XGBoost (XGB) algorithm [111].

For training and evaluating the models, we used the corresponding algorithms implemented in the Python `scikit-learn` [112] and `XGBoost` [111] libraries. Each model was trained on three different sets of predictor variables, listed in Table 2.6.

We utilised the relevant techniques implemented in the Python `scikit-learn` [112] and `XGBoost` [111] libraries to train and evaluate the models. Table 2.6 shows the three separate sets of predictor variables where each model was trained on.

Hyper-parameter tuning and model comparison

To discover the optimum hyper-parameters for each model, a grid search with 5-fold cross-validation was performed, maximising the Area Under the Curve (AUC) for classification tasks and the MAE for regression tasks. The AUC is calculated by plotting the Receiver Operating Characteristic (ROC) curve (graphing the real positive rate versus the false negative rate at various thresholds) and calculating the area below the curve. When comparing actual and predicted values, the MAE score is defined as the average of the absolute values of the errors.

Tables 2.7, 2.8 and 2.9 present the hyper-parameter grid for all models utilising the entire set of features in each of the four tasks. A grid search was performed for Logistic Regression (LR) and Elastic Net (ENet) using the standard classes `LogisticRegressionCV` and `ElasticNetCV`, with a `l1_ratio` grid of [0.0, 0.2, 0.4, 0.6, 0.8, 1.0] and [0.1, 0.5, 0.9, 1.0] respectively. The final values selected by cross-validation were 1 and 0.9. The names of the hyper-parameters correspond to their names in the `scikit-learn` library [112]. All of the parameters not displayed here were left at their default settings (see the documentation for more details). Finally, Tables 2.10, 2.11 and 2.12 present the set of optimal hyper-parameters for each of the models.

Parameter	Classification	Regression
`n estimators`	[200, 500, 1000]	[100, 200, 500]
`max samples`	[None, 0.8]	[None, 0.8]
`max depth`	[None, 2, 4]	[None, 4, 8]
`min samples split`	[2, 8, 32]	[2, 8, 32]
`min samples leaf`	[1, 4, 16]	[1, 4, 16]
`max features`	[None, sqrt, log2]	[None, sqrt, log2]

Table 2.7: Grid of hyper-parameters employed in the RF models.

Parameter	Classification	Regression
`n estimators`	[50, 100, 200]	[400]
`learning rate`	[0.01, 0.1, 0.2]	[0.01, 0.1, 0.2]
`max depth`	[None, 3, 6]	[None, 3, 6]
`min samples split`	[2, 4, 8]	[2, 4, 8]
`min samples leaf`	[1, 2, 4]	[1, 2, 4]
`max features`	[None, sqrt, log2]	[None, sqrt, log2]

Table 2.8: Grid of hyper-parameters employed in the GB models.

Parameter	Classification	Regression
`n estimators`	[50]	[50, 100, 200]
`learning rate`	[0.2]	[0.01, 0.1, 0.2]
`max depth`	[2, 4]	[2, 4, 6]
`subsample`	[1.0]	[0.7, 1.0]
`colsample bytree`	[1.0]	[0.5, 1.0]

Table 2.9: Grid of hyper-parameters employed in the XGBoost models.

	Classification		Regression	
Parameter	Binary	Three class	Standard	Threshold
`n estimators`	1000	500	100	200
`max samples`	None	None	None	None
`max depth`	None	None	None	None
`min samples split`	2	8	8	2
`min samples leaf`	1	1	4	4
`max features`	sqrt	sqrt	None	None

Table 2.10: Best hyper-parameters for the RF models, using all features.

	Classification		Regression	
Parameter	Binary	Three class	Standard	Threshold
`n estimators`	200	200	400	400
`learning rate`	0.2	0.1	0.01	0.01
`max depth`	6	6	None	None
`min samples split`	8	2	2	4
`min samples leaf`	4	2	4	8
`max features`	log2	log2	auto	auto

Table 2.11: Best hyper-parameters for the GB models, using all features.

	Classification		Regression	
Parameter	Binary	Three class	Standard	Threshold
`n estimators`	50	50	200	100
`learning rate`	0.2	0.2	0.01	0.1
`max depth`	4	4	6	6
`subsample`	1.0	1.0	0.7	1.0
`colsample bytree`	1.0	1.0	0.5	0.5

Table 2.12: Best hyper-parameters for the XGBoost models, using all features.

We report the F_1-score for binary classification as the harmonic mean of precision and recall assuming that the positive class is $y \geq 10\text{t}$. For multi-class problem, we report the average F_1-score weighted by the proportion of observations in each class.

2.3.4. Best model performance

Finally, we conducted a thorough examination of the best models for each task. In addition to the metrics stated above, we estimated the confusion matrix for the binary and multi-class classification problems in these analyses.

We also present the errors stratified by event type (set or deployment) and ocean basin to help evaluate whether the models are doing badly in particular subsets of the data. We confined these studies to binary classification and regression problems. Additional measures were produced, including F_1-score and accuracy for classification (see Appendix 2.A); and MAE and Symmetric mean absolute percentage error (SMAPE) for regression (see Appendix 2.B).

2.4. Results

2.4.1. Echo-sounder window selection

	Classification (F_1-score)		Regression (MAE)	
Hours	Binary	Three class	Standard	Threshold
24	0.911	0.811	10.16	8.70
48	0.919	0.813	10.05	8.63
72	**0.925**	**0.824**	**10.03**	**8.54**

Table 2.13: Model score for GB regression and classification models based on echo-sounder window size.

All GB models, regardless of task, displayed improved scores when using larger echo-sounder windows. The binary classification model (F_1-score = 0.925, Table 2.13) produced the best overall results throughout the classification tasks while employing the 72h echo-sounder window. Similar findings are exhibited for the regression tasks, with the 72h echo-sounder window producing the lowest MAE (Table 2.13). The threshold regression model performed better than the standard one, with a MAE almost 1.5t lower (Table 2.13). Because echo-sounder windows spanning 72 hours produced the best results across all models, this was the echo-sounder window used in the subsequent studies.

2.4.2. Classification models comparison

	Binary			Three class		
Models	Echo	Echo + Ocean	All	Echo	Echo + Ocean	All
Baseline	0.754	-	-	0.648	-	-
LR	0.885	0.889	0.895	0.773	0.788	0.799
RF	0.893	0.911	0.918	0.794	0.799	0.807
XGB	0.900	0.913	0.922	0.798	0.805	0.813
GB	0.907	0.924	**0.925**	0.791	0.812	**0.824**

Table 2.14: Classification on test events, F_1-score.

Table 2.14 shows the performance of all classification models evaluated for the 72h echo-sounder window. GB was the top performing model in both classification tests. Every model's performance improved as the number of features included in the training rose, i.e. as the models could learn from a bigger collection of features. Thus, the binary classification GB model trained with all features earned the greatest overall accuracy score (F_1-score = 0.925, Table 2.14). The ternary classification baseline model produced the least accurate results, being about 20% less accurate than the highest performing model for this test, the GB model with all characteristics. It is worth noting that the F_1-score increased between each ML model and the baseline.

2.4.3. Regression models comparison

	Regression			Regression (Threshold)		
Models	Echo	Echo + Ocean	All	Echo	Echo + Ocean	All
Baseline	12.85	-	-	11.40	-	-
ENet	13.99	13.70	13.52	12.18	11.84	11.60
RF	10.74	10.30	10.20	9.42	8.93	8.84
XGB	11.37	10.86	10.76	9.60	9.13	9.02
GB	10.51	10.10	**10.03**	9.18	8.74	**8.54**

Table 2.15: Regression on test events, MAE (t).

In Table 2.15, the results achieved by all the models trained on the various sets of predictor variables during the 72h echo-sounder window are displayed. The GB model outperformed all other models in the classification tasks. More particular, the threshold regression GB model was the most accurate, with an MAE almost 3t lower than the baseline model for the identical task, and 1.49t lower than the regular regression GB model. It is also worth noting that, in terms of classification tasks, all models profited from the addition of location and oceanographic data, and were able to use this knowledge to enhance their predictions over models that were simply given echo-sounder data. Although some of these differences were minor and maybe insignificant, it was evident that the ML models enhanced the baseline (which was the best of multiple feasible aggregations, Section 2.3.3) and benefited from adding all variables.

2.4.4. Best models results

When we examine the confusion matrix for the test set of both classification tasks (Figure 2.6), we notice that the GB model had a good success rate in classifying whether tuna biomass was < 10t or ≥ 10t, with just 6.03% of instances misclassifying. The ternary classification GB model, on the other hand, found it more difficult to distinguish between $10\text{t} \leq y < 30\text{t}$ and $y \geq 30\text{t}$ biomass predictions, misclassifying results in these two classes in 11.14% of cases.

	Sets		Deployments		All	
Ocean	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy
Atlantic	0.876	0.787	0.991	0.982	0.869	0.878
Indian	0.953	0.911	0.982	0.965	0.939	0.935
Pacific	0.964	0.930	0.997	0.994	0.953	0.980
All	0.934	0.878	0.991	0.983	0.925	0.938

Table 2.16: Errors for the binary classification task using the best model (GB), by ocean and event type.

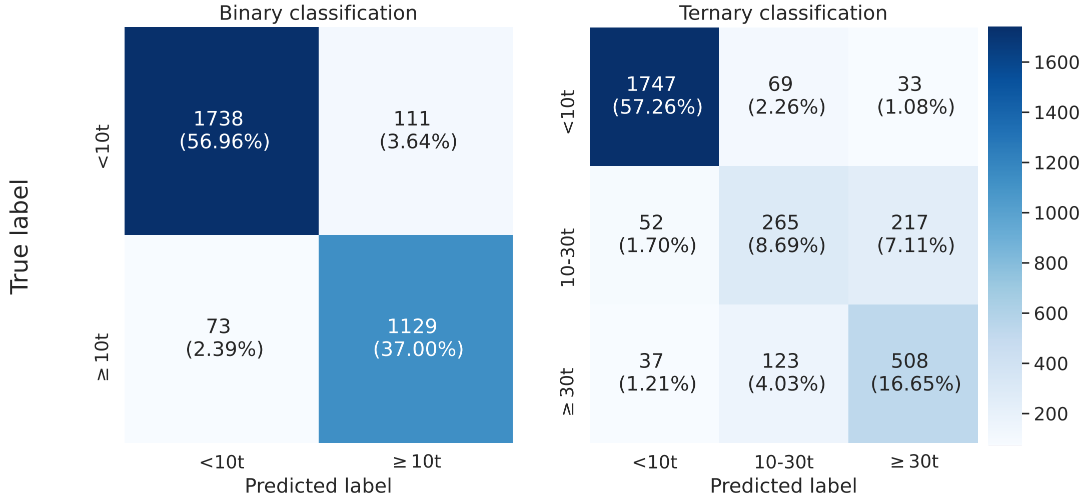


Figure 2.6: Confusion matrices with the best classification model performance on the test set. True label refers to the actual biomass category, whereas predicted label is the category inferred by the model.

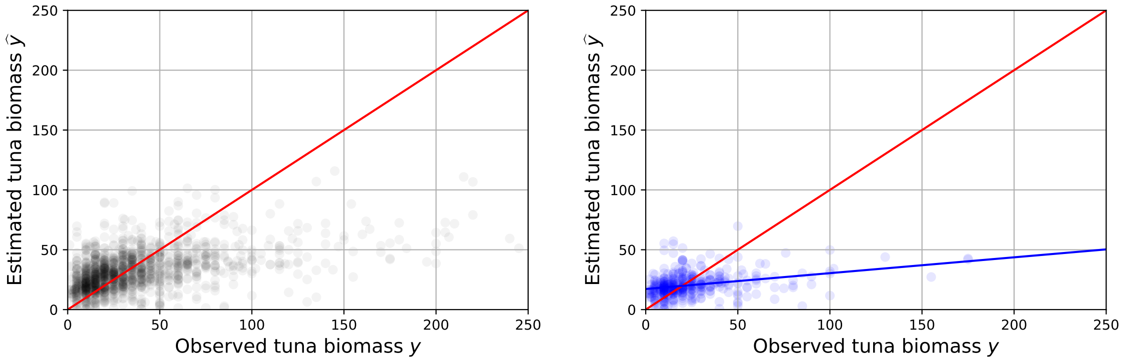
The findings of the binary classification model are shown in Table 2.16 by ocean basin and event type. Here, we can observe that the model performed better when tested on deployments (Accuracy = 0.983) than on sets (Accuracy = 0.878), i.e., it performed better when tuna biomass was < 10t rather than ≥ 10t. It is also worth mentioning that the Atlantic Ocean saw a significant decline in accuracy when compared to the rest, especially when tested on sets.

	Sets		Deployments		All	
Ocean	MAE (t)	SMAPE (%)	MAE (t)	SMAPE (%)	MAE (t)	SMAPE (%)
Atlantic	14.40	30.05	2.95	92.92	9.07	59.33
Indian	23.57	29.55	1.72	52.25	13.84	40.99
Pacific	27.96	28.52	0.36	32.09	6.44	31.31
All	21.66	29.51	1.29	51.15	10.03	41.86

Table 2.17: Regression task errors when using the best model (GB) by ocean and event type.

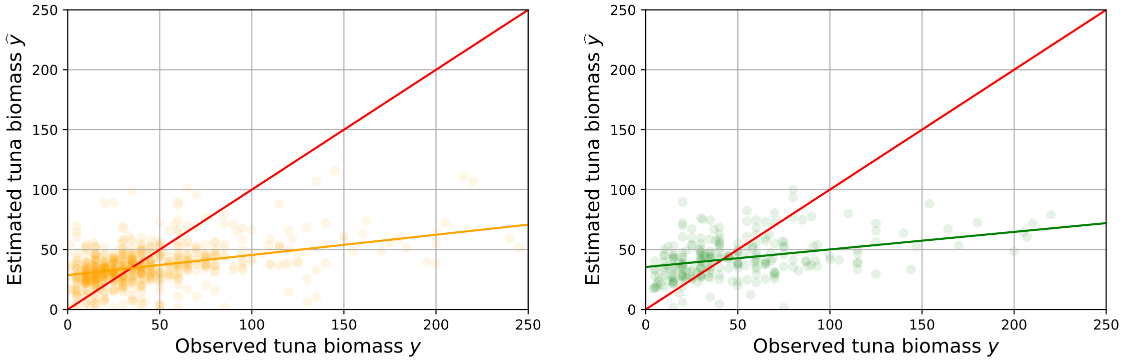
The MAE shown in Table 2.15 hides an important fact: the errors were very different for the two events included in the test data. Indeed, deployment events have by definition an observed biomass of zero: when tested over deployment events, the GB model had a MAE of 1.29t, while the MAE for set events was 21.66t (see Table 2.17). The reported overall MAE of 10.03t is thus the weighted average of these different populations.

The MAE in Table 2.15 reveals a key fact: the errors for the two events included in the test data were considerably different. Indeed, deployment events had an observed biomass of zero by definition: when evaluated across deployment events, the GB model had an MAE of 1.29t, but the MAE for set events was 21.66t (see Table 2.17). Thus, the stated total MAE of 10.03t is the weighted average of these different populations.



(a) Total

(b) Atlantic Ocean

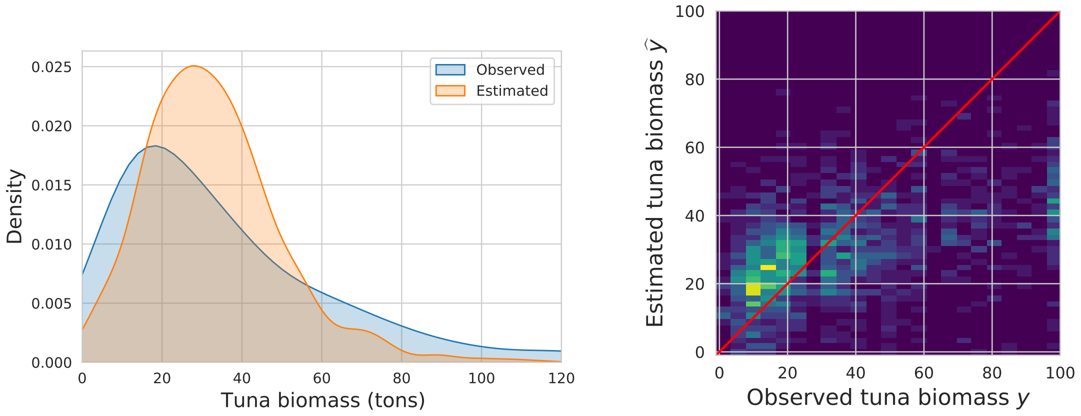


(c) Indian Ocean

(d) Pacific Ocean

Figure 2.7: Scatter plot of the observed against estimated tuna biomass in set events by ocean. The red line represents $\hat{y} = y$ and is shown for reference.

When looking more closely at the predictions of the best regression model, as shown in Figure 2.7a, it becomes clear that the model systematically underestimates cases of extremely high tuna biomass ($y \geq 100$t). This finding is consistent with the previously indicated improvement of the threshold regression task in comparison to standard regression. The MAE over set events was reduced to 18.33t for this model, and it was similarly decreased over deployments to 1.18t. Even with this threshold, however, the model tended to underestimate when observed tuna biomass was large (Figure 2.8b).



(a) Density distributions (set events) of observed and estimated tuna biomass for the standard regression task.

(b) Two-dimensional histogram of observed and estimated tuna biomass for the regression threshold task plot.

Figure 2.8: Error distribution of the two regression tasks

Some possible explanations for this underestimate are discussed in Chapter 5 (Section 5.1.1). There were no significant variations in performance between ocean basins (see 2.7). Figure 2.8a depicts the marginal distributions for observed and estimated tuna biomass on set events.

2.4.5. Feature importance

The interpretability of ML models is a difficult task: understanding which features the model deems most important in its computations. There are numerous techniques to assessing feature significance, and in this study we use permutation importance [92], which measures the significance of a particular feature as the loss in model performance when the values of that column are randomly shuffled in the training set.

We rate the ten most essential features for each task in Table 2.18 for the best model (GB). We next explain what each feature means:

- `Baseline` is the tons of tuna estimated by the baseline model, which aggregates the values of the 72 × 10 echo-sounder window.
- `N_Zero` is the number of zero-readings in the echo-sounder window, which can range between 0 and 72. If the overall biomass estimation is less than 1t, the buoy does not communicate an hourly biomass estimate to the satellite.
- `Max.LY` is calculated by taking the greatest of the 72 values for each layer `Y`, with layer 1 being the closest to the surface.
- `Max.HX` is calculated by taking the greatest of the 10 values for each hour `X`, with hour 0 being the closest to the event.

- `O2.DX` and `SSHa.DX` are the dissolved oxygen and SSHa, respectively, for each day `X`, with 0 being the closest to the event.
- `Latitude` and `Longitude` are the buoy locations closest in time to the event.
- `Ocean` is a category variable indicating the ocean basin where the event occurred.

	Classification		Regression	
Rank	Binary	Ternary	Standard	Threshold
1	`Max.L5`	`Max.L5`	`Baseline`	`Baseline`
2	`Longitude`	`Longitude`	`Max.L5`	`Max.L5`
3	`Ocean`	`Ocean`	`Longitude`	`Longitude`
4	`N_Zero`	`Max.L2`	`Latitude`	`Max.L2`
5	`Latitude`	`Max.H10`	`Max.H32`	`SSHa.D0`
6	`Max.L7`	`SSHa.D0`	`Max.L2`	`Latitude`
7	`Max.L1`	`Latitude`	`SSHa.D0`	`Max.L6`
8	`Max.L6`	`SSHa.D3`	`Max.H35`	`SSHa.D1`
9	`Baseline`	`O2.D1`	`SSHa.D1`	`SSHa.D3`
10	`Max.L2`	`O2.D0`	`SSHa.D3`	`Max.H10`

Table 2.18: For each of the four tasks, we display the top ten most significant features for the GB model. These variables are coloured according to the feature group to which they belong. Blue represents echo-sounder features, green is for oceanic variables, and red represents geographical coordinates (or features derived from them).

The interpretation of feature importance must be done with caution because there are definitely correlation between the variables, which must be taken into consideration when assessing permutation importance. However, we may briefly analyse the presence of some of them in Table 2.18, albeit more research is needed to confirm their veracity. For example, in both regression tasks, the feature `Baseline` emerges as the most important explanatory variable, which is understandable given that it is a first-stage biomass estimator. `N_Zero`, on the other hand, appears as a meaningful predictor solely in the binary classification task, because a reading of < 1t presumably means that there is no tuna beneath the DFAD. Furthermore, the geographical coordinates `Latitude` and `Longitude` show as important covariates for each task, emphasising the value of including this information source in the models. We also see that `Max.L5` consistently appears as one of the most relevant features for every job, which might be proof of the common depth at which tuna are present. Finally, several oceanographic factors, such as SSHa and `SSHa`, appear as important features in the three most demanding tasks, indicating that they assist the models estimate tuna tonnes more correctly.

2.5. Discussion

The TUN-AI pipeline was presented in this chapter, which uses echo-sounder buoy, FAD logbook, and oceanographic data to generate ML models that can reliably estimate tuna biomass aggregated at DFADs. The efficiency of several classification and regression models, as well as the impact of diverse data sources on model performance, were investigated. In this section, we will compare our findings to earlier research on estimating tuna biomass at DFADs, which was introduced in Section 2.2.

The methodology we presented varies from earlier studies in key respects. Although Baidai, Dagorn, Amande *et al.* [57] approach is comparable to ours (see Section 2.2.1), they only address the classification problem and hence cannot directly estimate the metric tonnes of tuna under the DFAD. They also have a lesser sample size in terms of sets (albeit similar) that only span the Atlantic and Indian seas. Finally, we evaluated various models for each task to see which one provided the highest overall performance. Despite the differences in technique and dataset utilised in the current study and the studies provided in Table 2.1, we believe it is important to incorporate TUN-AI results in the new enlarged Table 2.19 for context.

Ocean	Model	N	Accuracy	Recall	Specificity	F_1-score
Atlantic	EN [64]	3159	0.34	-	-	-
	RF [91]	22 501	-	0.79	0.99	-
	SVM [91]	22 501	-	0.79	0.99	-
	MLP [91]	22 501	-	0.77	0.99	-
	ID3 [91]	22 501	-	0.73	0.99	-
	IBK [91]	22 501	-	0.75	0.98	-
	RF [57]	1856	0.76	0.83	0.67	0.78
	RF [97] [a]	838	0.66	0.74	0.59	0.67
	TUN-AI	2869	0.88	0.88	0.88	0.87
Indian	EN [64]	26 143	0.47	-	-	-
	RF [57]	13 671	0.85	0.81	0.90	0.84
	RF [97] [a]	2144	0.58	0.63	0.53	0.60
	TUN-AI	4926	0.94	0.96	0.90	0.94
Pacific	**TUN-AI**	4400	0.98	0.98	0.98	0.95

[a] Predicts by-catch presence during fishing sets for tuna.

Table 2.19: Table 2.1 (approaches to predict tuna presence seen in the literature), with our results included.

Other studies that address the regression problem, such as Lopez, Moreno, Boyra *et al.* [60] and Orue, Lopez, Moreno *et al.* [99] (see Section 2.2.2), cannot be directly compared with this study for a variety of reasons. To begin with, their sample sizes are orders of magnitude less (21 and 287 sets, respectively). Second, they only have information from one ocean (Atlantic and Indian, respectively). Finally, they fit a statistical model, whereas our work uses an ML approach with a train-test split and a significantly bigger dataset. This means that TUN-AI is expected to have the stated performance on new, unknown data, however there is no assurance that the models in Lopez, Moreno, Boyra *et al.* [60] and Orue, Lopez, Moreno *et al.* [99] will generalise as well, given they use the same dataset for model fit and error assessment.

Furthermore, other work's assumptions and data-processing methods may not be exactly comparable to the approach detailed here. Lopez, Moreno, Boyra *et al.* [60] and Orue, Lopez, Moreno *et al.* [99], for example, assume that tuna exclusively inhabit levels deeper than 25m, eliminating biomass estimates from shallower layers out of their calculations. In our situation, all layers were taken into account since skipjack tuna prefer warmer surface waters where the thermocline is shallow [113]. In fact, subsequent research that used the same method as Lopez, Moreno, Boyra *et al.* [60] did not obtain substantial improvements in biomass estimations [99]. Baidai, Dagorn, Amande *et al.* [57] decided to use all layers in their studies while constructing tuna presence/absence and classification models, which utilised data from a different brand of echo-sounder buoys in the Atlantic and Indian seas but did not consider oceanographic characteristics in their models. To offer a full summary of the present level of tuna biomass estimation, we believe it is still necessary to incorporate TUN-AI results in Table 2.20.

Ocean	**Model**	**N**	**R^2**	**MAE**
Atlantic	Manufacturer [60]	21	0.25	21.29
	GLM [60]	21	0.73	-
	POL2 [60]	21	0.81	-
	POL3 [60]	21	0.83	10.25
	GAM [60]	21	0.82	-
	TUN-AI	2869	0.31	9.07
Indian	Manufacturer [99]	287	0.02	-
	GLM [99]	287	0.02	-
	POL2 [99]	287	0.02	-
	POL3 [99]	287	0.03	-
	GAM [99]	287	0.03	-
	TUN-AI	4926	0.38	13.84
Pacific	SEAPODYM [100]	300	0.85	-
	TUN-AI	4400	0.53	6.44

Table 2.20: Table 2.2 (approaches to estimate tuna biomass seen in the literature), with our results included.

Our study also looked at how oceanographic conditions and position-derived factors affected model performance. When compared to the model that just employed echo-sounder data, the incorporation of extra features dramatically improved across all tasks and models. This emphasises the significance of including contextual information into biomass estimations when utilising data from echo-sounder buoys attached to DFADs. Although this appears to be a laborious job at first glance, the current pipeline relies on automated processes for extracting oceanographic data and relating it to other available datasets, so the added complexity translated to only a few minutes of additional computation time on standard equipment. Given the increase in model accuracy when this information is included, as well as the potential benefits of having reliable techniques for estimating tuna biomass at DFADs, we believe it is desirable to utilise all available information. Previous research has looked at the association between tropical tuna distribution and oceanographic conditions using both catch data from observer logbooks and DFAD data. Skipjack tuna, for example, has been seen to congregate near upwelling systems, where feeding environment is favourable, and factors such as sea surface temperature or SSH have been demonstrated to have a substantial relationship with tuna distribution [63], [64]. Furthermore, Spanish fishermen that use echo-sounder buoys on DFADs believe that the oceanographic context of the DFAD, as well as the features of each ocean, impact the accuracy of biomass estimations produced by buoys [40].

Readers wanting to read further on the discussion of Tun-AI results are encouraged to skip to Chapter 2 (section 5.1.1). There we address any remaining questions that may remain after this chapter and suggest possible directions for future research to address these issues. Next chapter will show an application of Tun-AI pipeline to study tuna dynamics, with a focus on estimating how much time do tuna schools spend associated to DFADs.

APPENDICES

2.A. Classification metrics

We will explain the most common metrics used for classification problems, and particularly in tuna presence studies. One data sample is defined as “positive” in there was tuna present for that observation, and “negative” when tuna was considered to be absent. Thus, the predictions of the model will belong to one of these four categories:

- **True Positive (TP)**: when the model correctly predicts tuna presence.
- **False Positive (FP)**: when the model predicts tuna presence, but tuna was absent.
- **False Negative (FN)**: when the model predicts tuna absence, but tuna was present.
- **True Negative (TN)**: when the model correctly predicts tuna absence.

Accuracy refers to the fraction of predictions made by the model that are correct. The closer its value is to 1, the better the model is.

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

Recall or sensitivity is another metric used to evaluate the performance of a classification model. It measures the proportion of positive cases (presence) that the model correctly identified as such.

$$\text{Recall} = \frac{TP}{TP + FN}$$

In contrast, the **specificity** is the proportion of negative cases (absence) that the model correctly identified as such.

$$\text{Specificity} = \frac{TN}{FP + TN}$$

Finally, **$\mathbf{F_1}$-score** is calculated as the harmonic mean of precision and recall, where precision is the proportion of positive cases that the model correctly identified as such. The F_1-score is a single value that represents the overall performance of the model.

$$\text{F}_1\text{-score} = \frac{2TP}{2TP + FP + FN}$$

2.B. Regression metrics

The following metrics are typically used to quantify the efficiency of regression algorithms. First is the **coefficient of determination ($\mathbf{R}^2$)**, a measure of the goodness of fit of a regression model. In this context, R^2 describes the extent to which the variance of the dependent variable (tuna biomass) can be explained by the independent variable(s) in a regression model - i.e. the acoustic signal or the oceanography data. It is calculated as:

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2},$$

where i represents a sample in the data set, N is the number of samples, and $\bar{y}$ is the average catch of the whole data set. For each i, y_i is its associated real catch and $\hat{y}_i$ is the predicted biomass for that sample (in the same units as y_i). R^2 ranges from 0 (bad fit) to 1 (good fit).

Next metric is the **Mean absolute error (MAE)**, a measure of prediction error in a set of predictions. It is computed as:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

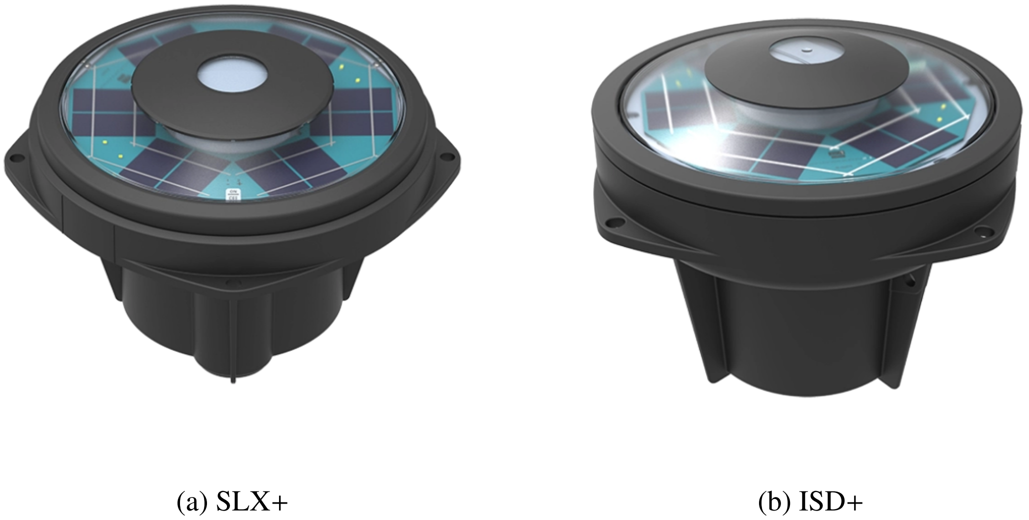
Our study also computes the SMAPE, defined as follows,

$$\mathrm{SMAPE}(\%) = \frac{100\%}{n} * \sum_{i=1}^{n}\frac{|\widehat{y_i} - y_i|}{|y_i| + |\widehat{y_i}|}$$

where y_i is the actual value and $\widehat{y_i}$ the estimated value. The main reason for reporting the SMAPE instead of the more common Mean Absolute Percentage Error (MAPE) is that the latter is undefined when $y_i = 0$, which happens with all the deployments. Besides, the SMAPE definition ranges between 0% and 100%, which makes it easier to interpret.

2.C. Echo-sounder buoys

The present study was made feasible by data received from Satlink's echo-sounder buoys, which were designed and deployed. Figure 2.9 depicts the look of these buoys. The following technical data was obtained from Lopez, Moreno, Boyra *et al.* [60]. The buoy is equipped with a Simrad ES12 echo-sounder, which runs at 190.5 kHz with a power of 140 W (beam angle at -3dB: 20°). The sounder is set to operate for 40 seconds. During this time, the transducer sends 32 pings, and an average of the back-scattered acoustic response is computed and saved in the memory of the buoy's software programme (hereinafter referred to as a "acoustic sample"). Volume back-scattering strength [114] values less than -45 dB are automatically deleted by the buoy's internal module as a precautionary step to exclude signals that are likely to belong to animals smaller than tuna [115], [116]. The observation depth range is 3 to 115 metres, with a blanking zone of 0 to 3 m. To reduce the transducer's near-field influence, data from this zone is eliminated [117]. Each observation is made up of ten homogenous layers with a resolution of 11.2 m. Because the echo-sounder buoys belonged to fishing crews, they were calibrated in tanks before to delivery rather than at the deployment location.


(a) SLX+

(b) ISD+

Figure 2.9: Two models of Satlink echo-sounder buoys used in this study. Images taken from their official webpage: `www.satlink.es`

Raw acoustic data are provided for each depth layer and were originally converted to biomass (in metric tons, t) by using an experimental algorithm developed by the manufacturer, which is based on the target strength of skipjack tuna, the main target species of the fleet fishing around DFADs. This conversion is automatically executed in the internal module of the buoy for each integrated layer by means of a depth layer echo-integration procedure [117], with the assumption that there was the presence of only individuals of skipjack tuna of identical weight.

Satlink's echo-sounder buoy takes one acoustic sample every five minutes, and sends one of this samples per hour to the database, that can be accessed by the fishing companies. An algorithm developed by Satlink chooses the acoustic sample to send, ensuring that the most relevant one is stored. Sampling frequency is lower at night for two reason. First, tuna is known to be less active at night. Second, the buoy is battery-free and stores solar energy in super capacitors. Doing less samplings at night ensures the buoy's stored energy lasts until the next day. On the other hand, sampling frequency increases at sunrise, as it is believed that tuna are more active around that time [60]. Satlink buoys also have an attached GPS, and stores their last known position in the database on a daily basis. Fishing vessels can increase the frequency of both acoustic samples and position data for a specific buoy on demand.

2.D. Code

The research presented in this chapter on sustainable fishing was developed using the Python programming language. Figure 2.10 provides a simplified representation of the flow diagram that was employed to process the data utilized in this study, highlighting the general behavior of the data processing pipeline. Furthermore, Figure 2.11 outlines the basic framework utilized for model training, providing a straightforward visual representation of the methodology employed.

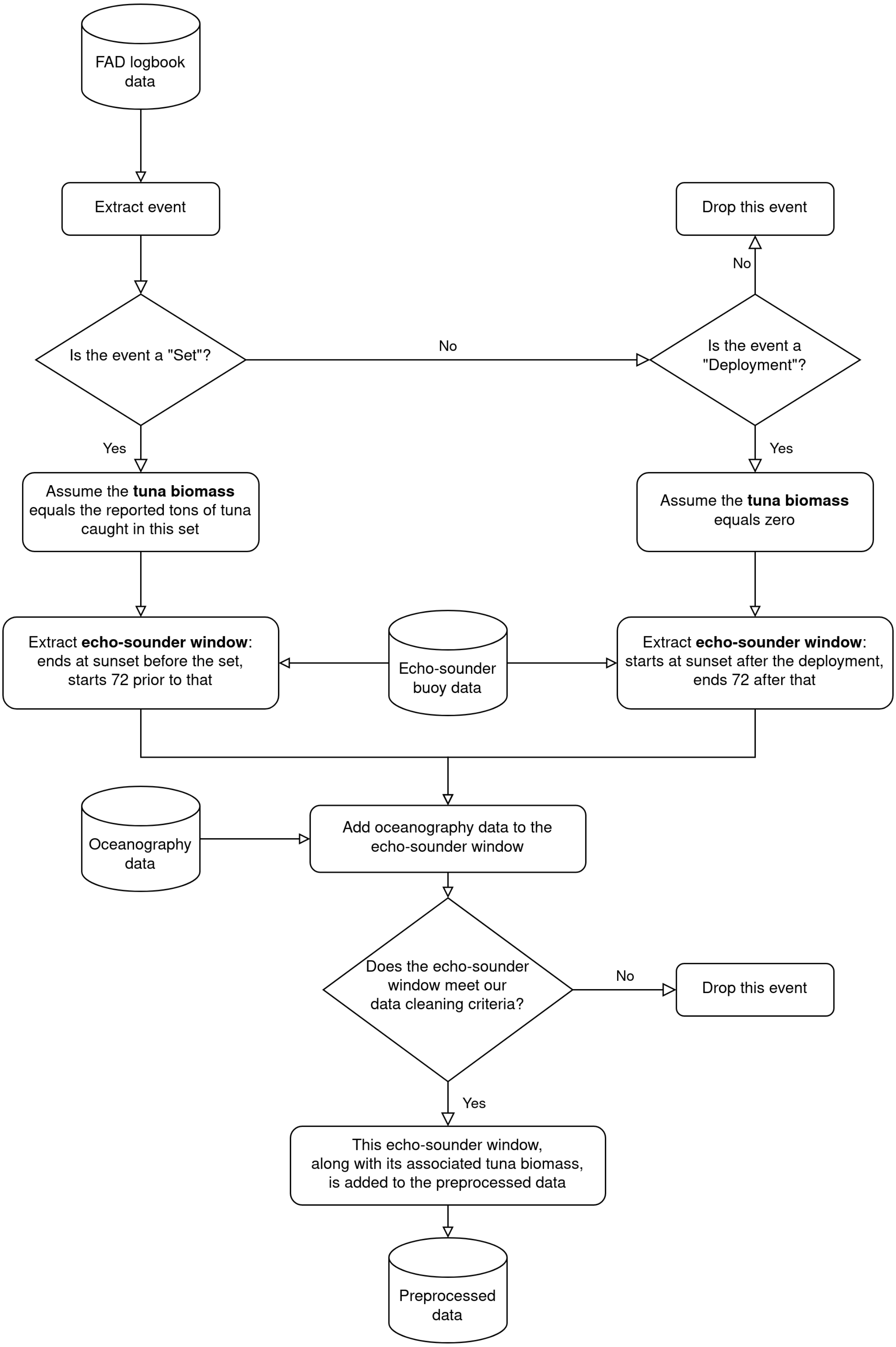


Figure 2.10: Basic representation of the flow diagram for the Tun-AI data processing, implemented in Python.

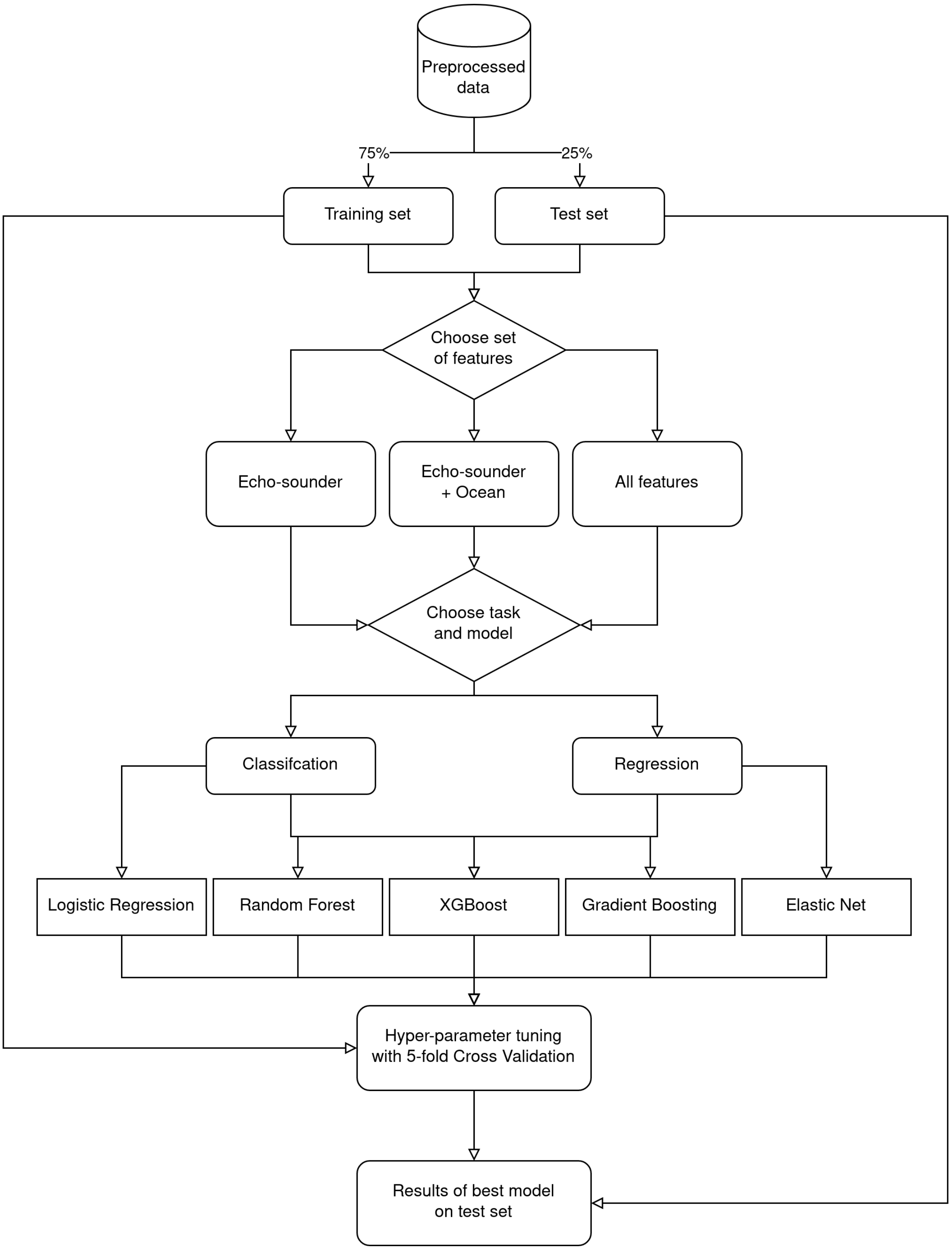


Figure 2.11: Basic representation of the flow diagram for the TUN-AI model training, implemented in the Python `scikit-learn` [112] and `XGBoost` [111] libraries.

3. TUNA DYNAMICS WITH TUN-AI

3.1. Introduction

The use of Drifting Fish Aggregating Devices (DFADs) in tropical tuna purse-seine fishing has sparked concern owing to the possible biological effects on tuna and the marine environment. As a result, there is a growing interest in understanding tuna dynamics and how they are influenced by DFADs. One solution is to employ the very own DFADs, which is equipped with echo-sounders that provide data on positioning and aggregated tuna biomass. In the previous Chapter 2 we presented Tun-AI , a Machine Learning (ML) pipeline that uses echo-sounder and oceanography data to estimate the biomass and presence of tuna under DFADs across tropical oceans. This chapter shows how Tun-AI can be applied to examine the temporal trends of tuna schools' association to drifting objects.

The structure of this chapter is as follows. First, the state of the art in tuna dynamics is introduced in section 3.2. Then, section 3.3 describes the data processing pipeline, particularly how biomass estimates are generated with Tun-AI and how these predictions are smoothed into time series. Results are shown in section 3.4. Using a binary output, metrics typically used in the literature are adapted to account for the fact that the entire tuna aggregation under the DFAD is considered. Using a regression output, two novel metrics, namely AT and DT, are estimated to obtain further insight into the symmetry of the aggregation process. These results are compared with the state of the art in section 3.5, where we conclude by addressing the "ecological trap" hypothesis.

3.2. State of the art

Given the concerns around the widespread use of DFADs in tuna fisheries today, it is not surprising that a considerable amount of research has been devoted to characterizing the dynamics at play when tunas aggregate to DFADs. However, results appear to be highly variable. One explanations lies in the inherent difficulties of conducting experiments in the open ocean. Most research on this subject is based on small-scale studies using electronic or acoustic tags to monitor individual tunas at a small number of DFADs. For these kind of studies, a small number of tuna individuals are caught close to a DFAD and surgically implanted a tag (see Figure 1.2), then released in the same area [118], [119]. DFADs used in these studies are equipped with special tools that detect the presence of tagged individuals nearby. Thus, researchers are able to compute the Continuous Residence Time (CRT) of tunas at DFADs, defined as the duration for which tuna was present at the FAD without day-scale absences [120]. The results for individual tagging studies are shown in Table 3.1.

Ocean	Study	Species	N	CRT (days)
Atlantic	Robert, Dagorn, Deneubourg *et al.* [121]	YFT[a]	24	4.05
		YFT[b]	16	1.65
	Tolotti, Forget, Capello *et al.* [119]	SKJ	7	9.19
		BET	23	25.31
		YFT	20	19.15
Indian	Dagorn, Pincock, Girard *et al.* [118]	SKJ	10	0.91
		BET	3	1.43
		YFT	55	1.04
	Govinden, Dagorn, Filmalter *et al.* [122]	SKJ	12	4.47
		BET	4	3.89
		YFT	13	9.98
Pacific	Matsumoto, Satoh and Toyonaga [123]	SKJ	15	2.30
	Matsumoto, Satoh, Semba *et al.* [124]	SKJ	30	1.30
		BET	32	3.80
		YFT	43	4.10
	Scutt Phillips, Peatman, Escalle *et al.* [125]	SKJ	13	1.00
		BET	97	10.00
		YFT	45	2.00

[a] Small individuals (30-39 cm).

[b] Large individuals (63-83 cm).

Table 3.1: Summary of main findings from previous studies on Continuous Residence Time (CRT) of individual tunas at DFADs. Species are: skipjack (SKJ), bigeye (BET) and yellowfin (YFT). “N” denotes the number of individuals tagged.

In Section 1.2, it was stated that certain assumptions need to be made in individual tagging studies, mainly that tagged fish experience the same processes and behave the same as untagged fish [34]. These assumptions are strong and should be addressed by tagging a substantial number of individuals, which is complex, expensive and potentially harmful for the tunas. An alternative to this is using the data collected from echo-sounder buoys to do study the aggregated population of tunas under the DFADs. As of today, this has the limitation of not distinguishing between tuna species, but the advantages outweight this shortcoming. Echo-sounder buoys are relatively cheap to deploy [40], can cover larger areas and are less invasive for fishes, as they do not require manual handling the individuals.

Studies on tuna dynamics using acoustic data are relatively new. Diallo, Baidai, Manocci *et al.* [126] computed CRT using echo-sounder data from two different buoy models (M3I and M3I+) drifting on the Indian Ocean in the period 2016-2018. They used a total of 5748 M3I buoys and 1368 M3I+ buoys. We show their computed CRT in Table 3.2, named Aggregation's Continuous Residence Time (aCRT) as it refers to groups of tuna rather than individuals. They also computed the Aggregation's Continuous Absence Time (aCAT), defined as the time between two consecutive associations to DFADs [121]; and the Colonization Time (CT), which is the time elapsed since the DFAD was deployed until tuna aggregated under it. All these metrics that characterize tuna aggregations will be explained further in Section 3.3.1.

Baidai, Dagorn, Amandé *et al.* [127] did a similar study to the previous one. Their research took place on the Atlantic and Indian Ocean from 2016 to 2018, and included 9118 trajectories of newly deployed DFADs. Along with providing the same metrics as Diallo, Baidai, Manocci *et al.* [126], these new study also computed the Soak Time (ST), which is the total number of consecutive days that the DFAD spend at sea; the Colonization Time (CT) of tuna on DFADs, i.e. the amount of time that takes for tuna to aggregate under a newly deployed DFAD; and the Occupancy Rate (OR), defined as the percentage of time that the buoy had aggregations under it. Refer to Section 3.3.1 for a broader explanation of these metrics.

Another study that uses echo-sounder data is Orue, Lopez, Moreno *et al.* [62]. In particular, their data comes from 962 echo-sounder buoys attached to DFADs, during three years of operations (2015-2018) in the Indian Ocean. The CT was computed, and the authors concluded that tuna takes on average 13.5 days to arrive at DFADs.

Summarizing all the previous studies, CRT has been found in the literature to range from less than a day to 55 days. Likewise, the values of Continuous Absence Time (CAT) ranges from 2 days to over 100 days.

3.2.1. Contribution

In this context, the current study applies the ML Learning based models from TUN-AI (described in Chapter 2) to provide accurate biomass estimates below DFADs across the Atlantic, Indian and Pacific Oceans, with the aim of characterizing the temporal patterns of tuna associations to DFADs. To do this, we adapt metrics already present in the literature to account for the fact that our study focuses on the entire tuna aggregation around the DFAD, as opposed to individual fish. Given that TUN-AI can deliver estimated amounts of tuna biomass aggregated to the DFAD, we examine the processes of aggregation and disaggregation in more detail. We check whether there could be a potential "ecological trap" [69], [70] effect on the tuna schools, by testing whether the time it takes for the tuna school to depart from the DFAD is significantly longer than the time it takes for the aggregation to form in the first place.

Metric	Ocean	Study	Mean	SD	Median
ST (days)	Atlantic	Baidai, Dagorn, Amandé *et al.* [127]	63.28	65.08	44
	Indian	Baidai, Dagorn, Amandé *et al.* [127] [a]	54.24	45.52	43
CT (days)	Atlantic	Baidai, Dagorn, Amandé *et al.* [127]	16.10	9.66	-
	Indian	Baidai, Dagorn, Amandé *et al.* [127]	40.46	17.31	-
		Diallo, Baidai, Manocci *et al.* [126] [a]	23.00	22.00	-
		Diallo, Baidai, Manocci *et al.* [126] [b]	17.70	18.00	-
		Orue, Lopez, Moreno *et al.* [62]	13.50	-	-
aCRT (days)	Atlantic	Baidai, Dagorn, Amandé *et al.* [127]	8.96	11.52	4
	Indian	Baidai, Dagorn, Amandé *et al.* [127]	6.20	6.86	4
		Diallo, Baidai, Manocci *et al.* [126] [a]	6.00	6.60	-
		Diallo, Baidai, Manocci *et al.* [126] [b]	8.40	10.90	-
aCAT (days)	Atlantic	Baidai, Dagorn, Amandé *et al.* [127]	5.38	6.01	4
	Indian	Baidai, Dagorn, Amandé *et al.* [127]	8.84	10.93	5
		Diallo, Baidai, Manocci *et al.* [126] [a]	9.70	12.00	-
		Diallo, Baidai, Manocci *et al.* [126] [b]	8.30	9.90	-
OR (%)	Atlantic	Baidai, Dagorn, Amandé *et al.* [127]	63.27	19.86	60.49
	Indian	Baidai, Dagorn, Amandé *et al.* [127]	45.45	46.16	21.73

[a] With buoy model M3I.

[b] With buoy model M3I+.

Table 3.2: Summary statistics, per ocean, for tuna aggregation metrics calculated in the literature. SD refers to the Standard Deviation.

3.3. Material and methods

3.3.1. Data processing

Data cleaning

Prior to analysis, the data must be cleaned of any records that may contaminate or obscure our study. To do this, a series of processes has been developed to eliminate potential errors:

- Duplicate rows and samples with missing buoy identification numbers are removed from the activity and echo-sounder databases, respectively.
- Echo-sounder data for sites less than 200m deep are eliminated because the echo-sounder signal may be influenced by the seafloor. This filter also excludes any land-based acoustic data.
- Acoustic measurements from buoys on board are removed by computing the mean buoy velocity over a day and deleting rows where the buoy speed exceeds 3 knots,

using the same criterion as Orue, Lopez, Moreno *et al.* [62].

Tun-AI estimates

A straightforward method for determining tuna biomass is to use solely raw acoustic data, as specified in Section 2.3.3. This baseline model uses the 72-hour echo-sounder window before the date of the prediction. The window takes one acoustic record each hour, and each acoustic record is made up of one value for each of the ten layer depths, resulting in a 72×10 matrix. Because the model's output (i.e. the total biomass estimation) is a single number, a set of aggregation criteria must be applied to the echo-sounder window matrix. The most accurate baseline model was the sum of all hours for each layer, followed by the mean of all layers. However, as we will see later, there are models that can outperform this baseline.

The biomass estimates provided by the echo-sounder may present variations when compared to real tuna tonnage under the DFAD [60]–[62]. This could be due to multiple causes, including the influence of oceanographic conditions or the diverse species composition under the DFAD. To mitigate this issue, we estimate tuna biomass using Tun-AI which has proven to be more accurate than simply considering the raw acoustic signal provided by the echo-sounder (i.e. the baseline model). Tun-AI , based on a GB algorithm [110] and trained using set and deployment events from the FAD logbook, uses information from the acoustic records, buoy location, and oceanographic variables to estimate the tuna biomass under DFADs. This pipeline includes:

The biomass estimations produced by the echo-sounder may differ from real tuna tonnage under the DFAD [60]–[62]. This might be attributed to a variety of factors, such as the impact of oceanographic conditions or the different species diversity beneath the DFAD. To address this issue, we estimate tuna biomass using Tun-AI , which has been shown to be more accurate than merely utilising the raw acoustic data supplied by the echo-sounder (i.e. the baseline model). Tun-AI , based on an GB algorithm [110] and trained using set and deployment events from the FAD logbook, estimates tuna biomass under DFADs utilising information from acoustic records, buoy position, and oceanographic factors. This pipeline is well explained in Chapter 2, but as a summary we list what Tun-AI includes:

1. a binary classification model trained to estimate whether the tuna biomass under a DFAD is higher or lower than 10t. Their scores are displayed in Table 2.16.
2. a regression model trained to give a direct estimate of the quantity of tuna biomass under a DFAD. This model scores are shown in Table 2.17.

Both models need a 72-hour echo-sounder window with one acoustic record per hour, and we constructed a baseline model that solely utilised aggregations of these raw echo-sounder estimates to assess the efficacy of our approach. Tun-AI also provides a three-class categorization model, which will be ignored in this chapter. One of the differences between our study and earlier ones is that we can evaluate both the aggregation and disaggregation processes to DFADs using a regression model, which would not be feasible with a binary classification model.

When developing these models, two major assumptions are made: first, the total catch captured by the vessel and later recorded in the FAD logbooks corresponds to the entire tuna aggregation present at the DFAD; and second, the entire tuna school is sampled by the echo-sounder beam at any point of the echo-sounder window. Such hypotheses are unavoidable in large scale research, and we anticipate that no substantial changes in the relative variation of the biomass will be noticed when the smoothing techniques outlined in Section 3.3.1 are carried out.

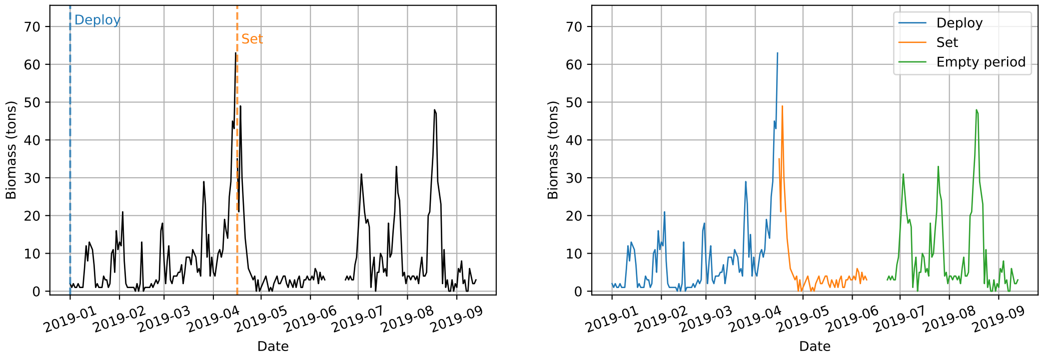
TUN-AI models may produce hourly biomass estimates for each buoy, however this frequency is insufficient for our investigation because of the noise created by day-night oscillations in tuna biomass [61]. To avoid this issue, we calculate daily biomass estimates for each buoy, totalling 3 873 531 outputs once the cleaning method mentioned in Section 3.3.1 is completed.

Generating unaltered segments

To eliminate the impacts of potential human interactions when examining tuna aggregation dynamics under DFADs, we divided the time series of each buoy into smaller segments in which such processes were not influenced by any external activity, which we refer to as *unaltered segments*.

To produce unaltered segments for every given echo-sounder buoy, we first combine the TUN-AI estimates (for both the binary classification and regression models) with the activity database, using the buoy identification number as the primary key. Only deployments, sets, retrievals at sea, recoveries at port, and losses were regarded to be "segment generating", that is, they may directly alter the echo-sounder data and the biomass dynamics under the FAD. Because visits and modifications were anticipated to have no influence on aggregated tuna biomass or echo-sounder data, they were excluded from this research. Finally, a unaltered segment would be generated if the buoy did not provide any information for longer than 24 hours, as this may suggest that the buoy was turned off or was otherwise unusable.

We only looked at segments longer than 72 hours because that is the minimum duration of the window required by TUN-AI to estimate biomass. We also excluded segments when TUN-AI failed to estimate more than 80% of the overall segment length. This may occur for extremely short segments (not previously rejected because they are longer than 72 hours) or if oceanographic data are not accessible (for example, due to difficulties with the CMEMS platform or data resolution). Otherwise, missing values from TUN-AI were interpolated in the regression model and propagated based on the most recent valid estimate in the binary classification model. Finally, using the preprocessing indicated in Section 3.3.1 and the processes given here, 43 334 unaltered segments were created. Figure 3.1 depicts the process of producing unaltered segments.



(a) The estimated biomass of a sample buoy over time, together with documented activities from the FAD logbook, are displayed as dashed lines.

(b) The unaltered segments were created using the actions recorded in the FAD logbook and a time with no buoy recordings. Each colour denotes one of the resultant unaltered segments.

Figure 3.1: The process of producing unaltered segments using biomass estimations and recording FAD logbook actions on a sample echo-sounder buoy over time.

Smoothing the signal

The output of both the binary and regression Tun-AI models is more indicative of genuine tuna biomass than the raw buoy values. However, considerable noise remains in the data, most likely owing to small scale changes in tuna aggregations or the effect of other fish species around the DFAD. We smoothed the obtained series to capture broad patterns while removing tiny oscillations since the goal of this chapter is to find general trends in tuna aggregation processes.

For the binary series, isolated estimates of one class or another are smoothed using the previous day's data (Figure 3.2). This smoothing process altered 2.7% of the total binary data.

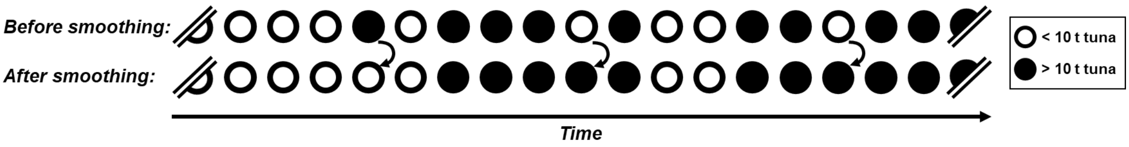


Figure 3.2: The smoothing technique for the binary series is depicted schematically: isolated estimations are adjusted based on neighbouring values.

In the regression model, we applied a constrained $P-$splines approach developed in Navarro-García [128], which captures the trend of the data without overestimating the signal while forcing the response to be non-negative (as the nature of the data requires). To smooth the series following this methodology, the open source Python package `cpsplines` is used [128]. Figure 3.3 shows the rightmost unaltered segment in Figure 3.1b together with its smoothed version.

We used a restricted $P-$splines technique established in Navarro-García, Precioso, Gavira-O'Neill *et al.* [4] in the regression model, which captures the trend of the data without overestimating the signal while constraining the response to be non-negative (as required by the nature of the data). The open source Python package `cpsplines` is used to smooth the series using this methods [128]. Figure 3.3 depicts the smoothed version of the rightmost unaltered section from Figure 3.1b.

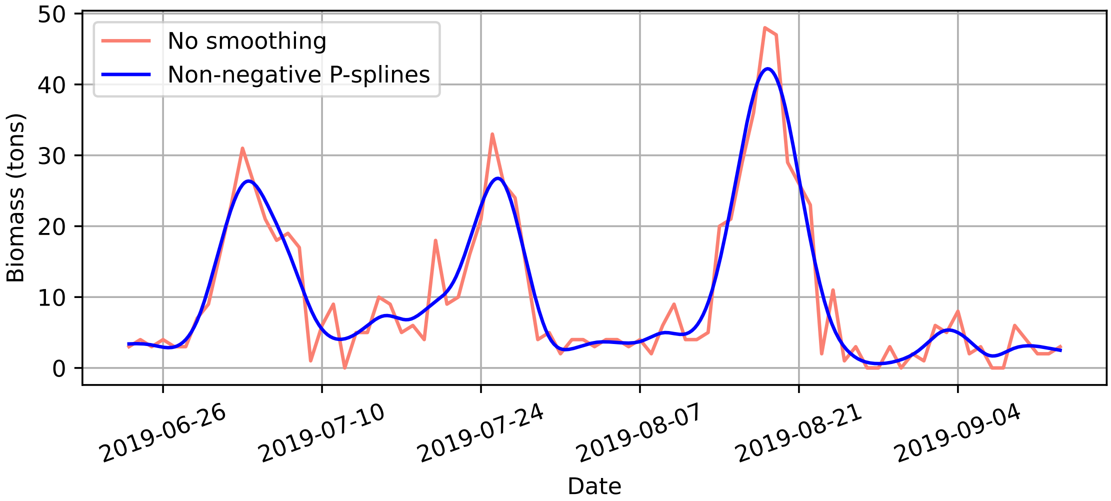


Figure 3.3: The smoothing technique for the regression series is illustrated above: The non-negative $P-$splines technique is used to smooth the biomass estimations provided by the TUN-AI regression model. TUN-AI predictions are in pink, while $P-$splines series are in blue. The smoothed curve is less affected by noise in the original data, allowing it to better capture overall trends while producing coherent estimates that meet the non-negative condition.

Tuna dynamics characterization

We estimate a variety of metrics utilising TUN-AI binary classification results for unaltered segments commencing with a deployment (7368 segments) to characterise the temporal patterns of the tuna school's aggregation to freshly deployed DFADs:

- **Soak Time (ST)**: represents the length of time a DFAD has been drifting at sea. Thus, the time elapsed from the first deployment of the DFAD till the end of the unaltered segment is computed here (Figure 3.4).
- **Colonization Time (CT)**: records the period from the initial deployment of the DFAD and the first detection of tuna [62]. Here, we quantify it as the duration

between the first deployment of the DFAD and the first day on which the binary model of TUN-AI produces a positive prediction, i.e. tuna biomass is larger than 10t (Figure 3.4).

- **Aggregation's Continuous Residence Time (aCRT)**: initially described by Ohta and Kakuma [120] as CRT for individually tagged tunas at DFADs and extended here to incorporate the complete aggregation, represents how long a tuna aggregation is continually detected by the echo-sounder buoy on a particular DFAD without day scale (> 24h) absences. aCRT is computed as the number of days in which TUN-AI has continually assessed tuna biomass higher than 10t (Figure 3.4).
- **Aggregation's Continuous Absence Time (aCAT)**: is derived also from Ohta and Kakuma [120]. This metric represents how long the tuna is missing from a given DFAD without day scale (> 24h) presences, and is determined here as the number of days when TUN-AI has consistently assessed tuna biomass that is less than or equal to 10t (Figure 3.4).
- **Occupancy Rate (OR)**: this variable is defined as the fraction of time that the tuna school remains at the DFAD following colonisation, and it may be approximated using the prior metrics.
- Percentage of DFADs that are never colonized: proportion of DFADs where the presence of tuna has never been observed. This helps to put the colonisation period statistics into context.

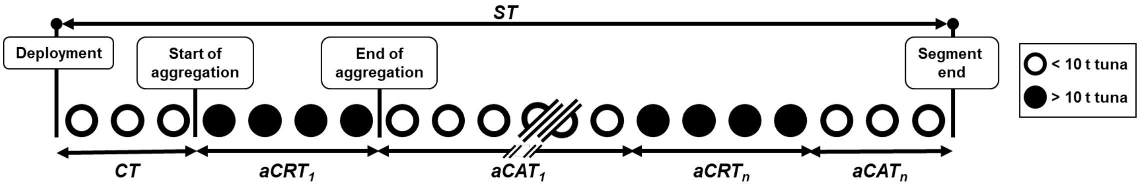


Figure 3.4: Schematic illustration of the results obtained by the TUN-AI binary model for a given DFAD, and the calculation of aggregation metrics based on unaltered segments starting with a deployment. Soak Time (ST); Colonization Time (CT); Aggregation's Continuous Residence Time (aCRT); Aggregation's Continuous Absence Time (aCAT).

Given that the current study also uses estimates of total tuna under the DFAD, the mechanisms of aggregation and disaggregation may be investigated. As a result, we propose two new metrics: Aggregation Time (AT) and Dissaggregation Time (DT). We use the daily tuna biomass estimates generated by the TUN-AI regression model after smoothing (see Section 3.3.1) to compute these. We identified the times when tuna biomass achieves a local maximum above 10t using these data, because this is the amount of tuna we consider to be a large aggregation. This was accomplished by modifying `scipy.signal.find_peaks` [129], and the peaks were identified by simply comparing neighbouring values of tuna biomass estimations. Peaks discovered within the first or last 5 days of the unaltered segment were deleted as an extra precaution to prevent the influence of human activity on biomass estimations. As a consequence, all unaltered segments lasting less than 10 days were removed, yielding a total of 23 326 unaltered segments. The total number of peaks is 71 644. AT was then computed for each peak as the time passed between the initial biomass estimate greater than 10t and the day when maximum biomass was attained. Similarly, DT was computed as the time between the greatest biomass estimate and the following biomass estimate under 10t. This procedure is depicted in Figure 3.5.

The binary classification Tun-AI model (see Section 2.4.2) uses a 10t cut-off for class discrimination, and we also employ this threshold in our investigation for numerous reasons. First, no major changes are expected because the models were trained across sets and only 7.6% of them reported less than 10t. Second, because smaller aggregations cannot be recognised due to model flaws, lowering this cut-off would not increase the quality of the estimations. Finally, while describing tuna dynamics under DFADs, we are primarily concerned with how the size of the tuna school evolves over time rather than its absolute value.

Kruskal-Wallis tests were used to see whether any of the previously described measures differed substantially among seas, and Dunn tests were used to validate pairwise differences. Similarly, Mann-Whitney tests were used to compare aCRT and aCAT, as well as AT and DT.

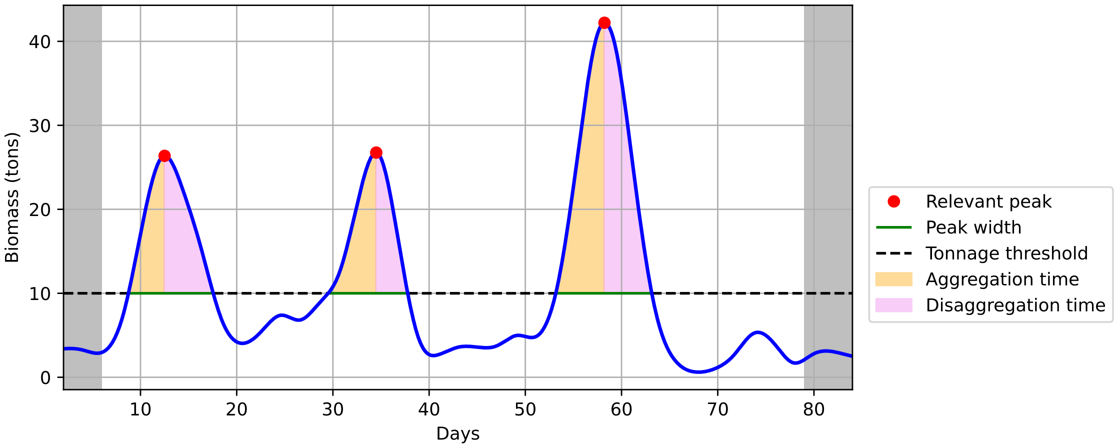


Figure 3.5: For a sample unaltered section, a schematic illustration of the computation of Aggregation Time (AT) and Dissaggregation Time (DT) using the smoothed biomass estimations provided by the Tun-AI regression model is shown. The shaded regions reflect days with no peaks, while the dashed line represents the 10t threshold.

3.4. Results

3.4.1. General aggregation metrics

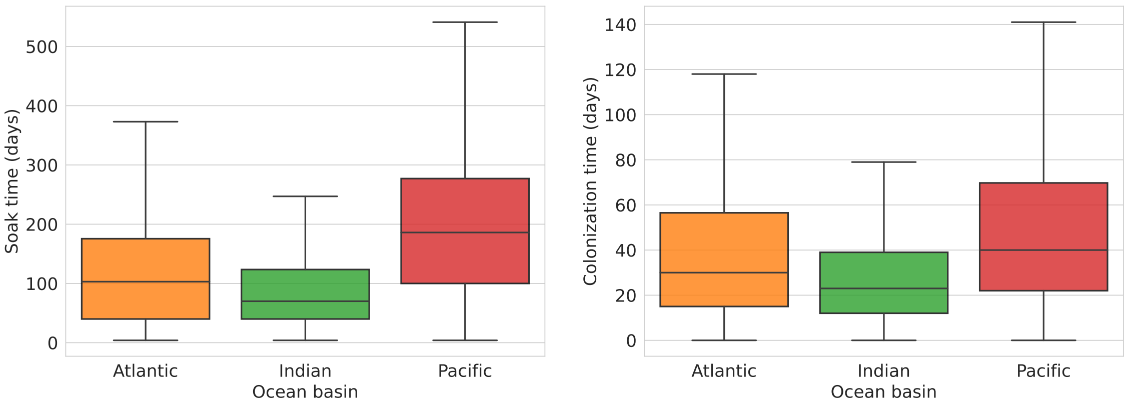
A overview of statistical metrics, categorised by ocean basin, is presented in Table 3.3 for the variables calculated from binary Tun-AI biomass estimations and utilising newly deployed DFAD, and their distributions are illustrated in Figure 3.6 through box plots.

Metric	Ocean	Count	Mean	SD	Median	IQR
ST (days)	Atlantic	1015	124	101	103	136
	Indian	1591	92	73	70	84
	Pacific	4762	202	129	186	177
CT (days)	Atlantic	1015	44	43	30	42
	Indian	1591	29	25	23	27
	Pacific	4762	51	42	40	48
aCRT (days)	Atlantic	3201	10	16	5	9
	Indian	4389	11	16	6	10
	Pacific	24 408	17	25	7	15
aCAT (days)	Atlantic	3875	24	33	11	24
	Indian	5088	19	23	11	21
	Pacific	26 552	21	30	9	21
OR (%)	Atlantic	1015	33	32	24	56
	Indian	1591	48	35	45	64
	Pacific	4762	53	31	55	48

Table 3.3: Summary data for tuna aggregation scores obtained from unaltered segments beginning with a deployment, per ocean. Includes SD and IQR.

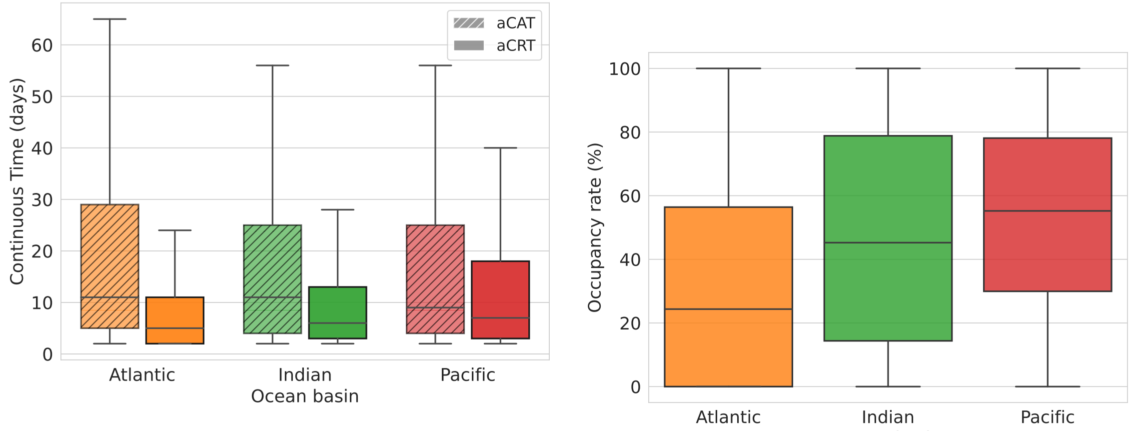
Regarding Soak Time (ST) and Colonization Time (CT), both show similar trends across oceans, as shown in Figures 3.6a and 3.6b. The Pacific Ocean has the longest CT and ST, while the Indian Ocean has the shortest, with the Atlantic Ocean falling somewhere in the middle. In fact, the Pacific Ocean's median ST is more than twice that of the Indian Ocean, while CT almost doubled it. In terms of variability, the Indian Ocean has the lowest standard deviation, whereas the findings for the other two oceans are more varied. The fraction of DFADs that were not colonised throughout their ST also varied significantly (27% in the Atlantic, 16% in the Indian, and 11% percent in the Pacific).

Looking at Figures 3.6c and 3.6d, we see that consistent patterns are apparent across seas for the Aggregation's Continuous Residence Time (aCRT), Aggregation's Continuous Absence Time (aCAT), and Occupancy Rate (OR), however trends diverge for the ST and CT. In this scenario, the Indian Ocean had aCRT, aCAT, and OR values that were between the Atlantic and Pacific Oceans. The Atlantic Ocean had the shortest times and the Pacific Ocean had the longest for the aCRT, whereas the converse was true



(a) Soak Time (ST)

(b) Colonization Time (CT)



(c) Aggregation continuous residence and absence time (aCRT and aCAT)

(d) Occupancy Rate (OR)

Figure 3.6: Box plots of the binary model's estimated variables. The inter-quartile rule of thumb with parameter 1.5 was used to eliminate outliers from the figure.

for the aCAT (Figure 3.6c). Hypothesis testing revealed significant differences across oceans for both aCRT and aCAT (Kruskal-Wallis test, $p < 0.01$), and these differences were verified in pairwise comparisons between oceans (Dunn test, $p < 0.01$). aCRT and aCAT median values throughout seas were typically comparable, ranging from 5 to 7 days or 9 to 11 days, respectively (Table 3.3). Overall, global aCRT was considerably lower than global aCAT (Mann-Whitney test, $p \gg 0.01$), and variability for aCAT was consistently larger than for aCRT. Finally, OR was about 50% globally, with the Atlantic Ocean having the lowest median OR at 24% (Table 3.3).

3.4.2. Aggregation and disaggregation times

We were able to analyse tuna aggregation dynamics around DFADs in greater depth using the Tun-AI regression model, calculating both Aggregation Time (AT) and Dissaggregation Time (DT). In general, AT and DT revealed comparable patterns throughout oceans, with the Indian Ocean having the shortest median AT and DT, and

		Aggregation time (days)				Disaggregation time (days)			
Ocean	Count	Mean	SD	Median	IQR	Mean	SD	Median	IQR
Atlantic	19 581	7.60	13.51	3.02	5.64	6.36	10.47	3.00	4.67
Indian	26 806	8.24	13.65	3.63	6.85	7.26	12.48	3.44	5.42
Pacific	25 257	15.45	25.04	5.89	13.99	14.63	25.99	5.49	11.03

Table 3.4: Tuna aggregation metrics summarised for the continuous model and decoupled by ocean basin.

the Pacific Ocean having the longest (Table 3.4, Figure 3.7). DT was not substantially longer than AT overall (Mann-Whitney test, $p \gg 0.01$). In fact, the first quartile for both AT and DT was typically identical, whereas the third quartile showed greater fluctuation, with AT generally longer than DT (Figure 3.7). Significance testing revealed variations in both AT and DT across oceans (Kruskal-Wallis test, $p < 0.01$) and across oceans (Dunn test, $p < 0.01$). Finally, regardless of the ocean where the DFAD was deployed, the distributions for AT and DT were positively skewed (the mean was bigger than the median).

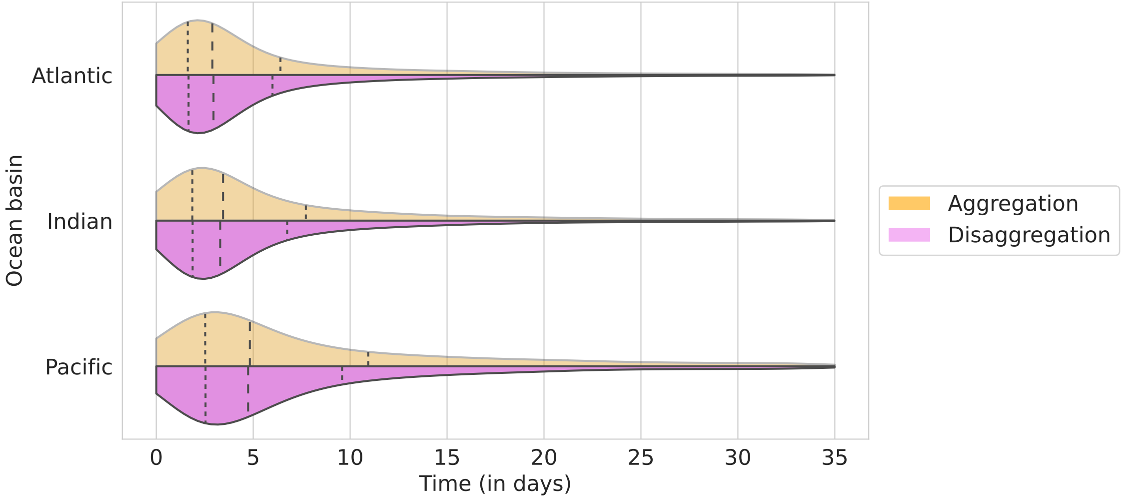


Figure 3.7: Violin plots illustrating the ocean basin specific aggregation and disaggregation time distributions. The vertical dashed lines represented the quartile positions.

3.5. Discussion

The study in this chapter intended to capture the overall trends in tuna aggregation dynamics at a global scale by employing data collected by echo-sounder buoys linked to DFADs over the period of many years and throughout all seas. This was accomplished using Tun-AI , a robust ML pipeline that analyses echo-sounder data to produce estimates of tuna tonnage beneath each DFAD as a binary output (< 10t or ≥ 10t) or as a direct estimate of biomass. To the best of our knowledge, this is the first study to look at standard tuna aggregation metrics (ST, CT, aCAT, aCRT, and OR) across all oceans and in such depth, offering insight into both the processes of aggregation and disaggregation in tuna colonisation.

Baidai, Dagorn, Amandé *et al.* [127] measured numerous parameters linked to tuna aggregation in the Atlantic and Indian Oceans using a binary model and a similar technique with echo-sounder buoys from a different manufacturer. In terms of Soak Time (ST), their estimations for the same seas are significantly shorter than ours (median values of 44 and 43 days in the Atlantic and Indian Oceans, respectively), most likely owing to definitional variations. While Baidai, Dagorn, Amandé *et al.* [127] defines ST as the "the number of days between the deployment of an DFAD equipped with a buoy and the first reported operation on it", our definition captures the length of the entire unaltered segment, which would likely be longer for buoys with no activities other than deployment, which made up 27% of our dataset. Escalle, Muller, Vidal *et al.* [130] reported mean drift periods of 118 days for DFADs included in the Parties to the Nauru Agreement's (PNA) FAD tracking trial programme in the Pacific Ocean, which is less than the median 202 days ST in our results. She does, however, point out that, due to data sharing limits, these timings are likely to be overestimated, as information outside of the PNA's Exclusive Economic Zones was not analysed. Fishermen have said that the typical lifespan of an artificial DFAD is around 5-12 months [63], which is consistent with the data recorded here.

In terms of Colonization Time (CT), there appears to be no widespread agreement among fishing masters. Moreno, Dagorn, Sancho *et al.* [131] interviewed fishing masters from the Indian Ocean, and around one-third of them believed that it takes at least one month for a school to aggregate to an DFAD. Although there is significant variety in the CTs reported in our study for DFADs across all seas, the median values are about 20-40 days, which corresponds to the observations of these fishing masters. However, approximately 45% of interviewed fishing masters believed that tuna colonisation of an DFAD was not time dependent [131], an observation that was also reflected in Lopez [63], where tuna abundance at DFADs was not positively correlated with ST, as evidenced by the large variability for CT in our data.

Although much research has been done on the time spent by tunas near and far from floating objects, the majority of work has been done on individually tagged tunas at a small number of study sites [107], [119], [120], [123], [124], [132]–[135]. These approaches give a high degree of detail, but they may not be reflective of underlying themes among all DFADs, or even of the overall patterns of a school of tuna. Robert, Dagorn, Deneubourg *et al.* [121], for example, discovered size dependent variations in the time yellowfin spent around an anchored FAD, with smaller individuals (< 50cm fork length) spending around four times as much time around the FAD as bigger individuals. Similarly, disparities in the Continuous Residence Time (CRT) of skipjack, yellowfin, and bigeye tuna between seas have been found [119], [132], [133].

Even while the huge data available from echo-sounder buoys coupled to DFADs may not give such fine-grained insight, it does show promise for discovering general trends in how large tuna aggregations behave. Diallo, Baidai, Manocci *et al.* [126], like us, utilised echo-sounder data from two DFAD buoy types from a different manufacturer to estimate Aggregation's Continuous Residence Time (aCRT) and Aggregation's Continuous Absence Time (aCAT) in the Indian Ocean. Both aCRT and aCAT were shorter than ours, and there were considerable disparities in buoy models ($6 - 8$ and $8 - 9$ days, depending on buoy model). This is a crucial consideration when comparing the findings of several research utilising echo-sounder buoys. Diallo, Baidai, Manocci *et al.* [126] concludes that the enhanced sensitivity of the newer model may be generating the discrepancies in aCRT and aCAT, therefore it stands to reason that buoys from various manufacturers will also report biomass differently. For example, the use of different frequency echo-sounders is expected to have an effect on the biomass estimations supplied by different buoy brands [40], [136]. Indeed, fishing masters observe variances in biomass estimates from different manufacturers [40], thus these differences should be handled with caution. Fishing technology advances fast, and researchers must keep up with manufacturers when drawing inferences from technology derived data. Nonetheless, even when comparing buoy brands and seas, aCRT and aCAT are often fewer than 10 days [126], [127]. This is consistent with the median and average CRT values discovered by most other authors when investigating individual tunas around DFADs (see Baidai, Dagorn, Amandé *et al.* [127]and references therein).

In a broader perspective, one of the primary concerns about DFAD deployment has been the prospect that DFAD might act as an "ecological trap", causing tuna to remain linked with the DFAD even as it wanders into areas that are not conducive to tuna growth and development [69], [70]. While other writers have analysed the current literature and determined that there is insufficient evidence to support or refute this concept [71], further study is needed. One of the novel aspects in our study was the application of a regression model to the echo-sounder buoy data, which allowed for direct estimates of tuna biomass aggregated to the DFAD (see Section 2.4.3), as well as the calculation of two derived metrics: Aggregation Time (AT) and Dissaggregation Time (DT), which could provide further insight into the nature of tuna's association to DFADs. Given that one of the grounds for an ecological trap is that the tuna's relationship with the DFAD is "rapid, strong, and long-lasting" [69], DT should be longer than AT if the DFAD were actually "catching" the tuna. However, our findings demonstrated that this was not the case, since DT was not considerably longer than AT. In reality, median AT and DT values did not reveal differences longer than a day, and when disparities were evident, such as in the third quartile, the time it took for the tuna aggregation to depart was actually faster than the time it took for the aggregation to form in the first place. Although these findings should be investigated further, there does not appear to be evidence of an ecological trap on a global scale.

In conclusion, TUN-AI proves to be a powerful tool to study population dynamics of tuna, and their results are aligned with findings in the literature. There is still further work that can be done, but we reserve that discussion to Chapter 5 (section 5.1.2). The last two chapters have focused on sustainable fishing, presenting the TUN-AI pipeline for tuna biomass estimation, developed in collaboration with Komorebi AI and Satlink. One of the novelties in this study was that it is the first to combine echo-sounder and oceanography data. Next chapter switches the focus to weather routing, describing another use for oceanography data: the optimization of maritime shipping routes.

APPENDICES

3.A. Code

The present research on tuna dynamics was conducted using the Python programming language. A simplified flow diagram outlining the data processing pipeline employed to generate the unaltered segments utilized in this chapter is presented in Figure 3.8. This diagram provides a high-level overview of the methodology used in this research, outlining the steps involved in the generation of unaltered segments, which served as the basis for subsequent analysis.

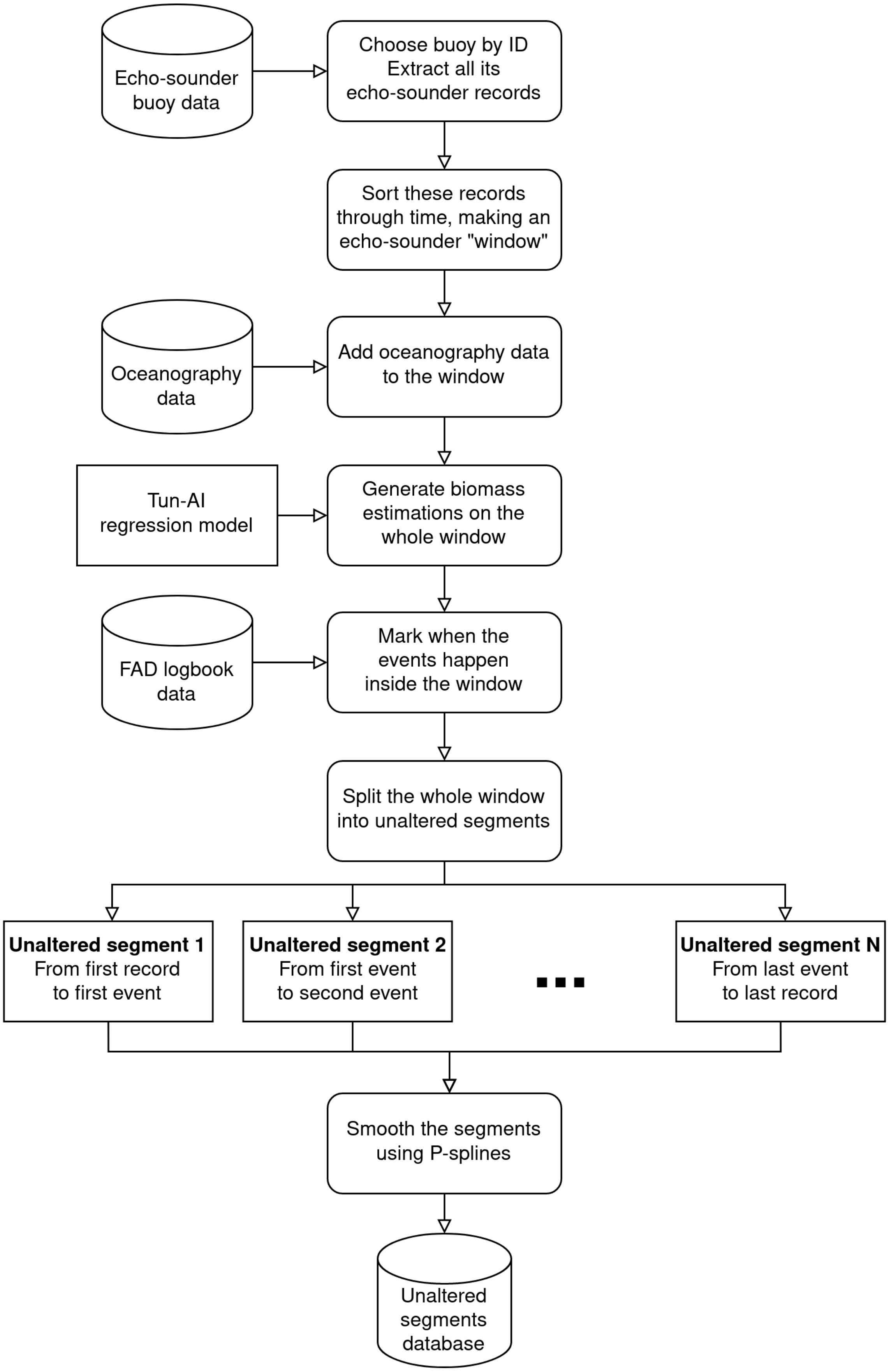


Figure 3.8: Basic representation of the flow diagram for the generation of unaltered segments using TUN-AI and the $P-$splines approach developed in Navarro-García [128].

4. WEATHER ROUTING

4.1. Introduction

In recent years, both academics and industry have paid close attention to ship weather routing. Problems in this field include determining the best sailing course and speed for a particular maritime journey, taking weather conditions into account. The goals are often to minimise operational expenses, fuel consumption, or passage danger. This chapter presents a study on weather routing, developing a system that combines real-time information with ocean context forecasts to plan the best route between two ports.

The structure of this chapter is as follows. First, the state of the art in weather routing is introduced in section 4.2. Then, section 4.3 describes the data sources available for this study and the design of benchmarks. Section 4.4 describes the optimization methods applied to those benchmarks. Results are shown in section 4.5 and we derive conclusions from them in section 4.6.

4.2. State of the art

Optimizing maritime routes under varying weather conditions is a complex mathematical problem. It can be addressed from different approaches by diverse study fields, from Mathematics to Compute Science. This section provides an overview of weather routing research, covering the major methodological approaches and disciplines involved.

One of the first approaches was the **isochrones method** [137], first applied by Hanssen and James [138] to optimise routes under stationary weather conditions. An isochrone is a set of connected points that a vessel may reach in a certain amount of time by departing from one point and travelling in all feasible directions, as shown in Figure 4.1. The points of an isochrone will depend on the vessel speed and its performance under the given ocean conditions. In order to reach the goal, one can construct several isochrones, chaining them so that the points of one isochrone serve as the starting coordinates for the next one. Several studies have developed and improved the isochrones method [139]–[141], and is up to this date one of the major used approaches [75]. One shortcoming of the isochrones method is that it relies on exploration i.e. simulating many possible paths to follow, and thus is computationally expensive, specially when taking into account time dependant weather conditions.

Another option for weather routing is utilising **variational optimization**, using a Lagrangian that can be optimized on a continuous space with gradient methods. Haltiner, Hamilton and Arnason [142] first proposed this approach to minimise time in a static setting where the speed relies on the wave height and direction. Another studies [143],

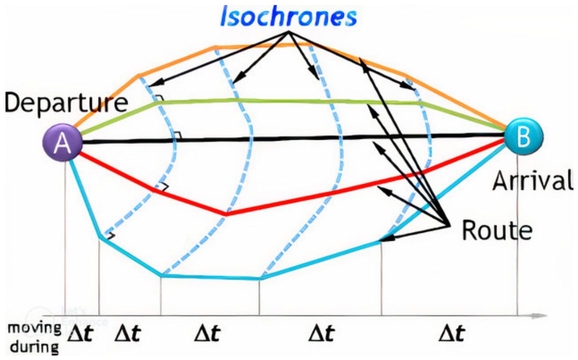


Figure 4.1: Overview of the isochrone method, from Roh [140].

[144] applied variational optimization methods on static ocean current vector fields to find locally optimal routes, reducing travel time. Variational optimization methods are exploitative: given an initial route, they change it to reduce any given cost function (fuel or time). As such, they are fast but rely on a good initial guess to obtain a optimal route.

A significant number of studies have utilised **dynamic programming** to enhance weather routing [75], [145]. This approach is based on Bellman's concept of optimality [146], which states that a problem can be divided into many simpler sub-problems, and that by finding each optimal sub-solution one gets the optimal solution of the original problem. In the context of weather routing, dynamic programming finds optimal routes by optimizing small segments and chaining them. The solution for each segment (sub-problem) can be done with any other optimization method, or via a custom set of rules [74], [147].

Graph optimization has also been applied to weather routing, traditionally via Dijkstra's search algorithm [148], [149]. This approach consists on building a discrete graph that defines the possible locations of the vessel on the ocean, named nodes. Each node is connected to its neighbours via edges, which have a weight associated to them, that represents the cost of travelling between those two nodes (can be the time that takes the journey, the fuel consumed, or both). Graph algorithms search for the path between two non-adjacent nodes that has the minimum total cost. Similarly to the isochrones method, graph search is computationally expensive, but can be speeded up by the use of heuristics, as is the case with $A^\star$ search algorithm [150]–[152].

Finally, some studies have approached weather routing by applying **genetic algorithms** [151], [153]–[155]. These methods are based on evolving a set of routes using small perturbations until they reach an optimal criterion. In contrast to variational optimization, that modifies the initial route using gradient methods, modifications done by genetic algorithms are random (with maybe a few rules to follow). This makes genetic algorithms a highly exploratory approach, but its random nature does not ensure optimality and is computationally expensive.

Regardless of the optimization approach utilised, most of the research in weather routing has one of two goals: reduce fuel consumption or sailing time, or occasionally both in multi-objective formulations. As some surveys on the field have pointed out [75], different studies tend to show a wide range of achieved savings when their suggested methodology is used. This makes a challenging task to compare results across different approaches. A standardization of what counts as a saving (either on fuel, time, or both) would be advantageous for the weather routing community. To aid future studies in this area, some benchmarking examples would be ideal, which are possible thanks to the increasing amount of weather data available from open sources.

4.2.1. Contribution

Following the observation made previously, the main contribution of this chapter is to build some reproducible benchmarks for weather routing. Historical meteorology and oceanography data from open sources will be used to define realistic benchmarks following commercial trading routes. Additionally, some synthetic vector fields will also be defined, helping to test the optimization algorithms on a controlled environment. We will also provide a simple ship model, easy to implement but realistic enough to assert the efficiency of any weather routing approach. In addition, our study will be one of the few to test different approaches for the same benchmarks. In particular, we will implement a variational optimization approach, a graph search method, and a genetic algorithm.

4.3. Material

In order to implement a weather routing system, it is important to know the meteorology and oceanography during the journey and how it affects the vessel's fuel consumption. Data of ocean currents, wind and waves can be obtained from predictive models done by providers like NOAA and Copernicus. Research on naval engineering should be able to yield the fuel consumption and emissions for each type of vessel under any shipping regime.

4.3.1. Meteorology and oceanography

This study uses weather data from two open data sources: Copernicus [156] and NOAA [157]. Both repositories contain all the required weather data with high resolution, and are based on the Ocean General Circulation Model (OGCM). Similar to weather forecast systems, an OGCM provides simulation via a complex system of differential equations that are run periodically with updated initial conditions based on certain measurements. Ocean currents are available on a grid of $1/12°$ in the Copernicus Marine Environment Monitoring Service (CMEMS) "Global Analysis and Forecast" model, which is updated daily and provides a 10 day forecast. Data on ocean waves can be obtained from the same CMEMS model, with the same resolution and update frequency, or from NOAA "Wavewatch III". Copernicus has APIs and a Python client to ease and automate data download, which are usually stored in NetCDF format. Wind data can also be obtained from a number of providers, such as NOAA or CMEMS. The specific data we used in this study is described in Table 4.1.

Data	Units	Resolution	Frequency	Provider
Current speed at ocean surface	m/s	1/12°	Daily	Copernicus
Wave height	m	0.50°	Hourly	NOAA
Wave speed	m/s	0.50°	Hourly	NOAA
Wind speed at ocean surface	m/s	0.25°	Hourly	NOAA

Table 4.1: Meteorology and oceanography data.

4.3.2. Benchmarks

In order to test our optimization approach, we must define a set of benchmarks, containing as many different scenarios as possible. For instance, presence of land between the two ports, strong opposing currents, or highly variable vector fields. Different scenarios can be easily simulated by the use of synthetic benchmarks, which we will comment later. We want, however, to include some real scenarios within our benchmarks. We do not know the best possible route for all benchmarks, but a good point of comparison is the circumnavigation, i.e. the route of minimum distance, also named geodesic when no land is present between the two points. In most realistic scenarios, the optimal routes are found around the geodesic, specially for higher vessel speeds.

Synthetic benchmarks

It is important to test algorithms first on synthetic benchmarks for several reasons. For starters, synthetic benchmarks provide a controlled and consistent environment for testing, allowing for more accurate and reliable results. This is particularly useful when evaluating the performance of an algorithm under different conditions, as synthetic benchmarks can be easily manipulated to simulate a wide range of scenarios. Secondly, synthetic benchmarks allow for the testing of algorithms without the need for real-world data, which can be expensive and time-consuming to obtain. This allows for faster and more cost-effective testing and evaluation of algorithms. Third, synthetic benchmarks can be used to test the robustness and reliability of algorithms. By introducing challenges and variations to the synthetic benchmark, it is possible to assess how well an algorithm can handle different situations and environments. This can provide valuable insights into the limitations and potential improvements of the algorithm.

A very simple benchmark we can define is a circular vector field, centred in (a, b). The currents spin clock-wise and have an increasing intensity (defined by s) the further one strays from the centre. This is summarized in the following equation:

$$W(x_1, x_2) = \langle s \cdot (x_2 - b), -s \cdot (x_1 - a) \rangle$$

Where (x_1, x_2) are local coordinates on a 2-D plane. In this work, we are using $(a, b) = (-3, -1)$ and a scale factor $s = 0.05$, small so that a vessel with unitary velocity can overcome currents near the centre. When testing optimizers we will ask them to develop a path traversing the centre: from $\boldsymbol{x_A} = (3, 2)$ to $\boldsymbol{x_B} = (-7, 2)$. Algorithms are expected to follow the direction of favourable currents.

The next synthetic benchmark we will test is the one appearing in Ferraro, de Diego and Almagro [144], called "four vortices". This vector field is defined by the following equation:

$$W(x_1, x_2) = s \cdot (-R_{2,2} - R_{4,4} - R_{2,5} + R_{5,1}),$$

where each vortex is expressed as

$$R_{a,b}(x_1, x_2) = \frac{1}{3\left((x_1 - a)^2 + (x_2 - b)^2\right) + 1} \begin{bmatrix} -(x_2 - b) \\ x_1 - a \end{bmatrix}.$$

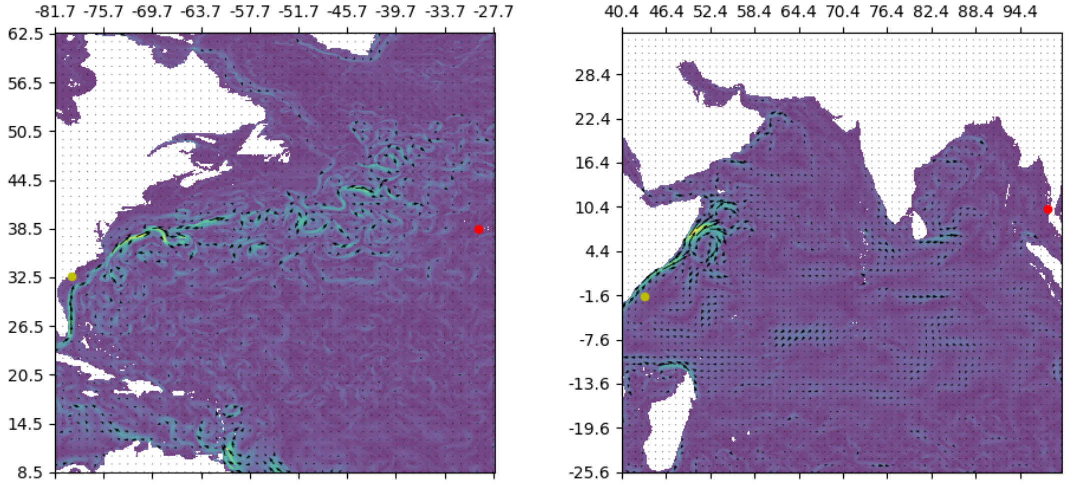
The authors explained that the scale factor $s = 1.7$ is chosen so that the maximum value of $|W|$ is almost 1. As we are testing these synthetic benchmarks using vessel with unitary velocities, we respect this factor. When testing optimizers we will ask them to develop a path from $\boldsymbol{x_A} = (0, 0)$ to $\boldsymbol{x_B} = (6, 2)$.

Real benchmarks

Oceanographic data for real case scenarios was downloaded from Copernicus Marine Environment Monitoring Service [156]. Copernicus has APIs and a Python client to ease and automate data download, which are usually stored in NetCDF format.

The first real example is a journey from Charleston (32.7°N 79.7°W) to Azores islands (38.5°N 29.5°W), traversing the Atlantic ocean during spring, the data starting specifically at the 25th of May of 2022. There is virtually no land in the trajectory. Ocean currents are relatively calm in this part of the Atlantic, but the route crosses the Gulf stream, a favourable current going north-east just at the departure point, see Figure 4.2a. The algorithm could take advantage of this current to save some time.

The last example is a journey from Somalia (1.66°S 42.39°E) to Myanmar (10.21°N 98.14°E), traversing the Indian ocean during summer, the data starting specifically at the 1st of July of 2022. There are several islands in between that the vessel needs to avoid in order to get to their destination, as shown in Figure 4.2b. Our goal when using this benchmark is precisely to test the ability of the algorithm to avoid land while also taking advantage of the vector field.



(a) Charleston-Azores

(b) Somalia-Myanmar

Figure 4.2: Real vector fields: one with only water (a) and one with pieces of land marked in white (b). Yellow point marks the starting position and red is the goal. The ocean currents are coloured by intensity, the fastest being represented with brighter (greener) colours.

4.3.3. Consumption models

The task of implementing a realistic consumption model is complex, because the use of fuel is affected by many different factors such as the resistance due to wind and waves, and the speed of ocean currents, among others. An even more difficult task is to adapt the model to each type of vessel: different kinds of ships will have distinct consumption models [158]. For this reason, there is a large body of literature involving modelling ship's consumption as a function of speed and weather conditions. Some of the main books include Hagiwara [139] and Barras [159].

The simplest theoretical consumption model is the **propeller's law**, which expresses a cubic relationship between speed and consumption. However, in real scenarios large deviations from this simple law are observed, and more specific models have been developed, which include advanced hydrodynamic modelling of the resistance due to waves, wind and swell and their effect on propulsion as a function of the ship's hull and characteristics. A recent review on this matter can be seen in Psaraftis and Kontovas [160], with an emphasis on slow steaming, a general reduction on vessel speed for marine transport during the 2009 crisis, which thanks to the non-linear relation between speed and consumption, was also responsible for considerable reduction in emissions.

The weather routing algorithms implemented in this study are designed to be flexible enough to use any consumption model. Due to the scarcity of vessel data at the point of writing this thesis, we were unable to design fine-tuned consumption models, so we resort to a general model. Even if it is not highly accurate, this consumption model will allow us to get a general idea on how effective our optimization algorithms are. In case we get more detailed vessel data in the future, we will be able to use accurate models that compute realistic estimations of the fuel consumption.

General models express the fuel consumption of a vessel (F, in grams per hour) as the product of its engine power (P, in kilowatts) and their Specific Fuel Oil Consumption (SFOC) (in grams per kilowatt-hour):

$$F = \mathrm{SFOC} \cdot P \tag{4.1}$$

SFOC can be expressed as a function of the engine load level, which is a percentage of the Maximum Continuous Rating (MCR) [161]. Most engines can be tuned to modify the SFOC curve, changing the optimal value where specific consumption is minimal [162]. When no information about the vessel engine is provided (as is our case), we use the standard SFOC value which is $185 g/kWh$ [161]. Fuel consumption is also affected by oceanography and weather. The power required by the vessel in presence of wind and waves can be modelled as the sum of power needed to move in calm water, plus the power required to overcome wave resistance, plus the force of wind on draught:

$$P = P_{\text{base}} + P_{\text{waves}} + P_{\text{wind}} \tag{4.2}$$

Ocean currents are easily incorporated into the model, as they only modify the speed of the vessel with respect to water. **Harvald's model** [163] expresses the power necessary to overcome the resistance of water as a function of the vessel speed over water (v_{water}, in meters per second):

$$P_{\text{base}} = \Delta^{2/3} \cdot v_{\text{water}}^3 / C \tag{4.3}$$

where Δ is the water displaced (the vessel's weight, in tonnes), and C depends on the vessel speed over water and its longitude (L, in meters):

$$C = 3.7 \left(\sqrt{L} + 75 / v_{\text{water}} \right) \tag{4.4}$$

In presence of waves, the vessel engine will need to provide extra power [164]. This is expressed as:

$$P_{\text{waves}} = \frac{1}{16} \cdot \rho_{\text{water}} \cdot g \cdot h_w^2 \cdot \sqrt{\frac{B^3}{L}} \cdot v_{\text{ground}} \tag{4.5}$$

where ρ_{water} is the water density (in kilograms per cubic meter), g is the acceleration due to gravity (in meter per quadratic second), h_w is the height of waves (in meters), B is the vessel's beam (in meters), and v_{ground} is the vessel speed with respect to ground (in meters per second). This speed takes into account the velocity of ocean currents, v_{currents} (in meters per second):

$$v_{\text{ground}} = v_{\text{water}} + v_{\text{currents}} \tag{4.6}$$

The force of wind on the vessel's air draught offers extra resistance or thrust depending on the wind direction. A simple model to take into account the wind resistance is:

$$P_{\text{wind}} = \frac{1}{2} \cdot K_X \cdot \rho_{\text{air}} \cdot A_X \cdot \cos(\varphi) \cdot v_{\text{wind}}^2 \cdot v_{\text{ground}} \tag{4.7}$$

where v_{wind} is the wind apparent speed in relation to the vessel (in meters per second), ρ_{air} is the air density (in kilograms per cubic meter), A_X is the area of the vessel's projection across its longitudinal axis (in quadratic meters), and K_X is a constant that depends on the type of vessel, with values usually ranging between -0.8 and -0.5. The angle of the wind with respect to the vessel, φ, is taken such as $\varphi = 0$ means the wind blows from stern to bow (favourable wind), which makes P_{wind} take a negative value because the wind is reducing the power needed by the engine.

If more data about the vessel is available, it is possible to use more accurate model, such as replacing Harvald's model for calm water by Holtrop's [165], or applying Liu's model for waves [166]. As we commented at the start of this section, the choice of one consumption model over the other does not affect the weather routing algorithms presented here, as the optimization explores the relative differences of consumption between ocean routes. The goal of this chapter is to prove that we can improve ocean routes via weather routing, reducing the fuel consumption given by any model.

This general consumption model has been tested against published data, and shows favourable results. In Figure 4.3 we show the estimations of our model compared to real data from Bialystocki and Konovessis [167]. The vessel used there was a PCTC (Pure Cars and Trucks Carrier) sailing through the Pacific Ocean. Our model predicts better around the design speed when we also consider the extra power due to wave resistance.

4.4. Optimization methods

We use algorithms that apply mathematical optimization, taking as objective function the model of fuel consumption in conjunction with oceanography and meteorology data. Different methods have been tested, including genetic algorithms, heuristic graph optimization, and variational methods with numeric integration.

4.4.1. Variational method

The first algorithm we introduce to perform weather routing is based on variational methods. We will model the route as a curve in space and time, and formulate the

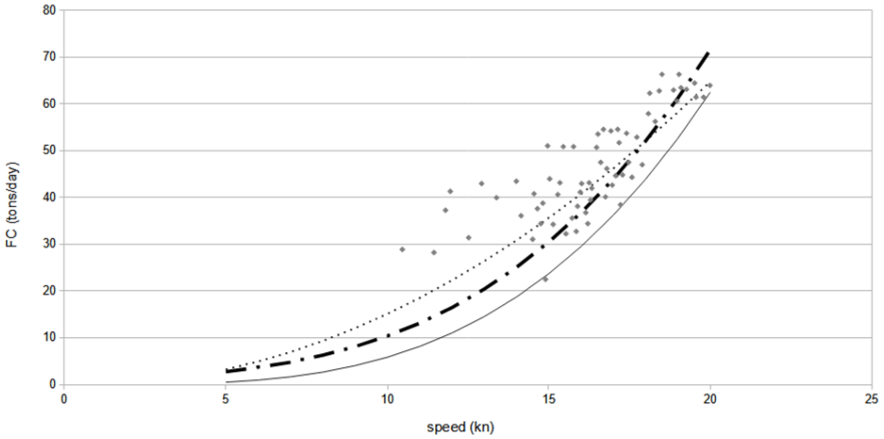


Figure 4.3: Fuel consumption curve of our model compared against data obtained during real operations. Dots: data from Bialystocki and Konovessis [167]; dotted line: second order interpolation from Bialystocki and Konovessis [167]; continuous line: our model without waves; dotted-slashed line: our model with waves.

optimization problem as finding the curve that minimizes a certain functional that captures the effects of the ocean. We are taking several assumptions to simplify the problem. First, we will only take into account the effects of ocean currents, i.e. the consumption given by equation (4.3). Second, although it is known that weather conditions change over time, we are assuming a stationary vector field for the sake of simplicity, though the variational method proposed is easy to adapt for evolving conditions. Third, we assume the ship keeps a constant velocity with respect to water. Lastly, real case scenarios have obstacles present, in the form of land. Our variational method is able to avoid small obstacles but it is not intended to perform optimal circumnavigation, i.e. bypass significant obstacles in the way to the goal.

To address the routing problem, we will use the classical formulation from Zermelo's Navigation Problem (ZNP). We first introduce its equations both on the plane and on the sphere, to then explain our variational method, named Hybrid Search (HS).

Zermelo's Navigation Problem on the plane

This problem was proposed in 1931 by Ernst Zermelo [168], is a classic time-optimal control problem, where its aim is to find **time minimum trajectories** under the influence of a drift vector $\vec{w}(x_1, x_2) = \langle w_1(x_1, x_2), w_2(x_1, x_2)\rangle$ where x_1, x_2 are local coordinates, and where w_1, w_2 are the vector components chosen relative to a local frame. This drift vector can be interpreted as wind or water current. In small scale simulations, the coordinates and the vector components can be taken to be Euclidean. Once we pass to larger scale simulations that take into account the Earths' curvature, the coordinates (x_1, x_2) indicate longitude and latitude delineated in degrees, while the vector components are taken relative to a local east-north framing and delineated in meters.

The goal is to navigate from a specified initial point along a path that minimizes time, under the influence of $\vec{w}$, assuming the vessel provides constant thrust V (speed over water) and has a heading angle (over water) α w.r.t. the x_1-axis. Thus, the velocity components over ground can be expressed as:

$$\begin{aligned} \frac{dx_1}{dt} &= V\cos\alpha + w_1(x_1, x_2) \\ \frac{dx_2}{dt} &= V\sin\alpha + w_2(x_1, x_2) \end{aligned} \tag{4.8}$$

Using the Calculus of Variations, one can show that such a path necessarily obeys the following differential equation, first derived by Zermelo [168]

$$\frac{d\alpha}{dt} = \sin^2(\alpha)\, w_{2,1} + \sin(\alpha)\cos(\alpha)\, \left(w_{1,1} - w_{2,2}\right) - \cos^2(\alpha)\, w_{1,2} \tag{4.9}$$

where for the sake of brevity, we write $w_{i,j} = \partial w_i / \partial x_j$.

Equation (4.9) is known as the **Zermelo differential equation**. Together with (4.8) it gives the form for time-optimal trajectories as a dynamical system in the 3-dimensional space parametrized by (x_1, x_2, α). For the sake of completeness, the derivation of this last equation is fully explained in Appendix 4.A.1.

Zermelo's Navigation Problem on the Sphere

We now modify the above derivations to the case where the ship is travelling on the surface of the Earth - idealized here as a perfect sphere. To that end, we adopt spherical coordinates $x_1 = \theta$ (longitude) and $x_2 = \phi$ (latitude) measured in units of κ radians. In particular, it may be convenient to take $\kappa = \pi/180$ if we wish to measure things using degrees. The background current will be given relative to a east-north framing, which we represent as the following 2×2 matrix

$$F(\theta, \phi) = \begin{bmatrix} K\cos\theta & 0 \\ 0 & K \end{bmatrix}$$

where K is the conversion scale from the units used to measure θ, ϕ and the units used to measure local velocities. For example, if global position is measured using degrees of arc, and local velocities are measured in kilometres, then letting R be the earth's radius in kilometres ($R \approx 6367$ km), we have $K = \kappa R = \pi R/180 \approx 111.1$ kilometres per 1 degree of arc.

With these conventions in place, the velocity over ground of a vehicle moving at a speed of V over water is given as

$$\begin{aligned} K\cos(\kappa\phi)\frac{d\theta}{dt} &= V\cos(\kappa\alpha) + w_1(\theta, \phi) \\ K\frac{d\phi}{dt} &= V\sin(\kappa\alpha) + w_2(\theta, \phi) \end{aligned} \tag{4.10}$$

where α is the ship's heading measured relative to an East-North framing, where $\vec{w}(\theta, \phi) = \langle w_1(\theta, \phi), w_2(\theta, \phi)\rangle$ with w_1 being the component of displacement relative to east, and w_2 the component of displacement relative to north.

Using the Calculus of Variations once again, now on the sphere, one can show that such a path necessarily obeys the following differential equation,

$$\begin{aligned}\kappa K\frac{d\alpha}{dt} &= \begin{bmatrix}\cos(\kappa\alpha) & \sin(\kappa\alpha)\end{bmatrix}\begin{bmatrix}\sec(\kappa\phi)w_{1,1} & w_{1,2}\\ \sec(\kappa\phi)w_{2,1} & w_{2,2}\end{bmatrix}\begin{bmatrix}\sin(\kappa\alpha)\\ -\cos(\kappa\alpha)\end{bmatrix}\\ &\quad - \cos(\kappa\alpha)\tan(\kappa\phi)(V + \cos(\kappa\alpha)w_1 + \sin(\kappa\alpha)w_2)\end{aligned} \tag{4.11}$$

Equation (4.11) can be justly considered the analogue of the Zermelo differential equation for motion on a sphere. For the sake of completeness, the derivation of this equation is fully explained in Appendix 4.A.2.

The Hybrid Search algorithm

Hybrid Search (HS) is a 3-step variational algorithm for solving the Zermelo-problem in either Euclidean or Spherical background. The 3 steps are (i) exploration, (ii) refinement, and (iii) smoothing. The output of the exploration and refinement phases is a piece-wise optimal trajectory that connects a starting location with a desired destination.

In effect, exploration is a shooting method based on the Zermelo Initial Value Problem (ZIVP). The exploration algorithm formulates multiple instances of a ZIVP with a given initial position and a fan of directions aimed towards the target. The trajectories are then evolved using RK4 numerical solutions to the Zermelo Differential Equation with dynamic termination conditions. The most obvious termination condition is to select the trajectory that minimizes the distance to the target. In practice, it turns out that a better heuristic is to terminate each trajectory when the difference between the heading angle and the direction to target exceeds a certain pre-set threshold. The algorithm is greedy, in that a single "winner" trajectory is selected from the list of dynamically terminated trajectories. This selection is performed on the basis of distance to target.

The refinement phase is just a refined version of the exploration algorithm, but this time the fan of initial directions is taken as small deviations from the winner trajectory of the exploration phases. The tightness of the refinement spread is constrained so that the fan of directions does not exceed the spread between two directions of the exploration phase. The candidate trajectories are then evolved using the same heuristic as in the exploration phase and a winner is selected based on proximity to the target. The precise details of the exploration and refinement sub-algorithms are detailed in the following subsections.

The third phase consists of smoothing the output of the refinement using the Ferrero-de Diego-Almagro Algorithm (FDA) algorithm. This algorithm is a numerical Boundary

Value Problem scheme that works by iteratively shifting a given discretized trajectory towards a time-minimizing route. The approach is based on the discrete Calculus of Variation and can, in principle, be utilized with any given Lagrangian. In our case, we select the time-minimizing Lagrangian such that the corresponding Euler-Lagrange equations are precisely the Zermelo Differential Equation. We then discretize the time-minimizing Lagrangian using a pre-selected time-step and begin the iteration with the piece-wise optimal solution generated by the exploration and refinement sub-algorithms. Because the initial trajectory is piece-wise optimal, the overall effect is that of smoothing the sharp turns present in the initial trajectory and converting the piece-wise smooth and piece-wise optimal solution to a smooth, near optimal solution of the Zermelo problem. The relevant details of the FDA algorithm are specified at the end of this section.

Exploration step

Given a starting point $\boldsymbol{x_A} = (x_{A,1}, x_{A,1})$, and a goal point $\boldsymbol{x_B} = (x_{B,1}, x_{B,2})$, we can first centre a "search cone" in the direction of $\Lambda_{A,B}$, following equation (4.12) (assuming an euclidean space). The amplitude for this cone is γ. If the vector field was null and we started a trajectory with heading $\alpha = \Lambda_{A,B}$, the vessel would eventually arrive to $\boldsymbol{x_B}$. Thus, by taking this search cone, we are assuming that the optimal route will always point close to the destination and that the vector field will have a small effect on the vessel trajectory. However, this assumption can be relaxed by increasing the amplitude of the cone, γ (up to 2π, covering all directions).

$$\Lambda_{i,j} = \Lambda(\boldsymbol{x_i}, \boldsymbol{x_j}) = \arctan\left(\frac{x_{j,2} - x_{i,2}}{x_{j,1} - x_{i,1}}\right) \tag{4.12}$$

Equation (4.12) defines the angle $\Lambda_{i,j}$ from point $\boldsymbol{x_i}$ to point $\boldsymbol{x_j}$. This equation is applicable in Euclidean space, and can be generalized to spherical geometry for short distances. However this does not hold for our study as distances between start and end points are significant, so when working in spherical space it is better to replace equation (4.12) by the following:

$$\Lambda_{i,j} = \arctan\left(\frac{-c_j \cdot s_i + c_i \cdot s_j}{-\left(c_i \cdot c_j + s_i \cdot s_j\right) \cdot \sin(x_{i,2}) + \left(c_i^2 + s_i^2\right) \cdot \sin(x_{j,2})}\right) \tag{4.13}$$

where $c = \cos(x_1) \cdot \cos(x_2)$; and $s = \sin(x_1) \cdot \cos(x_2)$.

Next, we generate N initial shooting angles, namely

$$\alpha_n(0) \in [\Lambda_{A,B} - \gamma/2,\ \Lambda_{A,B} + \gamma/2]\,.$$

To do so we N-sect the "search cone" into $\alpha_0, \ldots, \alpha_N$, evenly spread across the whole search cone, and use each of these as an initial condition to solve the system of ODE via the RK4, explained in Appendix 4.C. We will use these shooting angles to generate N local paths, or trajectories $q_n(t) = (x_{n,1}(t), x_{n,2}(t), \alpha_n(t)),\ n \in [0, N]$.

The N generated trajectories evolve using RK4, in iterations of time $\tau > \Delta t$ (where Δt is the time step of RK4). That means that, after iteration i, the routes will have evolved until time $t = i\tau$, and will be defined by the points $q_n(t),\ t \in [0, i\tau]$. We name the first iteration of this exploration step as i_0, which will start at $i_0 = 1$ but will be updated in further optimization steps. After every iteration i, each trajectory n is checked individually to assert whether it meets any one of three stopping conditions. If it does, trajectory n is left out of the RK4 loop and will not evolve further. These three rules are:

1. Trajectory n is stopped after iteration i if

   $$D\left(\boldsymbol{x_n}(i\tau), \boldsymbol{x_B}\right) \leq d,$$

   being $D(\boldsymbol{x_a}, \boldsymbol{x_b})$ the distance metric between two points, defined according to the space we are operating on, and d a certain distance threshold. This implies the vessel has reached its goal.

2. Trajectory n is stopped after iteration i if its heading $\alpha_n(i\tau)$ deviates too much from the goal. To assert this, we take point $\boldsymbol{x_n}(i\tau)$, and compute its angle to $\boldsymbol{x_B}$, named $\Lambda(\boldsymbol{x_n}(i\tau), \boldsymbol{x_B})$, see equations (4.12) and (4.13). Otherwise, the trajectory keeps evolving while the following condition is met:

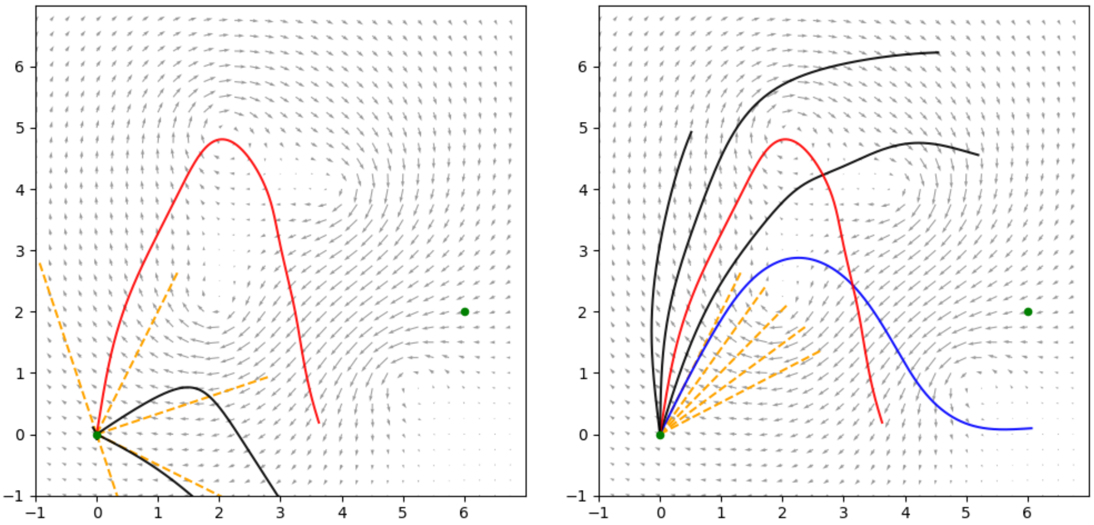
   $$(\Lambda(\boldsymbol{x_n}(i\tau), \boldsymbol{x_B}) - \gamma_d/2) \leq \alpha_n(i\tau) \leq (\Lambda(\boldsymbol{x_n}(i\tau), \boldsymbol{x_B}) + \gamma_d/2),$$

   where γ_d is the maximum deviation allowed from the goal, typically equal or lower than the search cone $\gamma_d \leq \gamma$. The higher γ_d, the more exploratory is this method, but it will take more iterations to converge.

3. Trajectory n is stopped after iteration i if any of its points $\boldsymbol{x_n}(t), t \in [0, i\tau]$ is located in land. In addition to stopping the trajectory, the algorithm discards all the waypoints $q_n(t),\ t \geq t_{\text{land}}$, being $\boldsymbol{x_n}(t_{\text{land}})$ the first point located in land. The trajectory $q_n(t),\ t < t_{\text{land}}$ is kept, as it may still be the optimal route and just needs a course correction, that will be done in a later step.

One can argue that rule 2 is too strict for small γ_d, as the vessel can be heading "wrongly" for a negligible amount of time before turning "correctly" again, and that the resulting route might be optimal. However, when working with real scenarios, the influence of the vector field is small enough to justify that a vessel going in a "wrong" direction won't turn "correctly" on time to compensate this deviation.

Figure 4.4a shows a visualization of this exploration step, highlighting the one which got closest to the goal. RK4 method ensures that all trajectories are time optimal. After



(a) (b)

Figure 4.4: First two steps of the Hybrid Search (HS) method: (a) exploration and (b) refinement. Each trajectory is generated from a different shooting angle (in orange) and evolves using Fourth order Runge-Kutta method (RK4) method iteratively with $\tau = 0.1$, until their heading deviates more than $\gamma_d = \pi/2$ radians from the goal. After all local paths are computed, the one that got closer to the destination is chosen as best (highlighted in the graph). The search cone had an amplitude of $\gamma = \pi$ radians in the exploration step and was centred on the direction of the goal. During refinement, the search cone was centred on the shooting angle of the best route found in the exploration step, and its amplitude is narrower, $\gamma = \pi/5$.

all N trajectories stop, if none of them reached the goal $\boldsymbol{x_B}$ (i.e. none met the 1st stopping rule), we keep the trajectory

$$m : D\left(\boldsymbol{x_m}(i_1\tau), \boldsymbol{x_B}\right) \leq D\left(\boldsymbol{x_n}(i_1\tau), \boldsymbol{x_B}\right) \forall n \in [0, N] \tag{4.14}$$

where i_1 was the last iteration from RK4 method. We named this trajectory m as our “best trajectory”, then move to the refinement step.

Refinement step

In the exploration step, we assumed that the optimal route should be heading closely towards the goal $\boldsymbol{x_B}$, and evolved trajectories defined by the points

$$q_n(t),\ n \in [0, N],\ t \in [0, i_1\tau].$$

with initial shooting angles $\alpha_n(0) \in [\Lambda_{A,B} - \gamma/2,\ \Lambda_{A,B} + \gamma/2]$.

We now generate a narrower search cone, with amplitude $\gamma_b << \gamma$ (for instance, $\gamma_b = \gamma/5$) and we centre it on α_m, where m is the “best trajectory” from the exploration step. Thus, the newly generated initial shooting angles are evenly spread across

$$\alpha_n(0) \in [\alpha_m - \gamma_b/2,\ \alpha_m + \gamma_b/2],\ n \in [0, N]$$

We now redo the exploration step, starting at iteration $i = i_0$. The first time we enter the refinement step, $i_0 = 1$ but that will be updated soon. The trajectories will stop eventually, at iteration i_2. Note that, as there are new trajectories, we now may take a different number of loops than the exploration step, and thus i_2 is not guaranteed to be equal to i_1. If no trajectory reached the goal $\boldsymbol{x_B}$ (i.e. no trajectory meets the first stopping criteria), we regenerate update the "best trajectory" m following equation (4.14). The algorithm goes back to the exploration step using $\boldsymbol{x_m}(i_2\tau)$ as the starting point, i.e. $\boldsymbol{x_A} = \boldsymbol{x_m}(i_2\tau)$; and starting at iteration $i_0 = i_2$.

This loop between exploration-refinement continues until the first stopping rule happens, i.e. one trajectory gets close enough to the destination $\boldsymbol{x_B}$. Figure 4.5a displays one possible result of this process. One issue is apparent: the vessel takes sharp turns in the connections between local paths. This happens because each trajectory (except the last one) is stopped due to deviating from the goal, so the vessel is forced to correct its course by turning sharply to reach its destination.

Smoothing step

We see that if we use the paths generated by our approach, it's not very "smooth", which is not realistic for real world situations. So we want to "smooth" it out while still optimizing the cost. Following Ferraro, de Diego and Almagro [144], we apply the Newton-Jacobi method to the discretized Euler-Lagrange equation to smooth out the path. Throughout this thesis we will refer to this algorithm as Ferrero-de Diego-Almagro Algorithm (FDA). Since each path is already local optimal, the FDA algorithm can converge to an optimal solution after a suitable number of iterations.

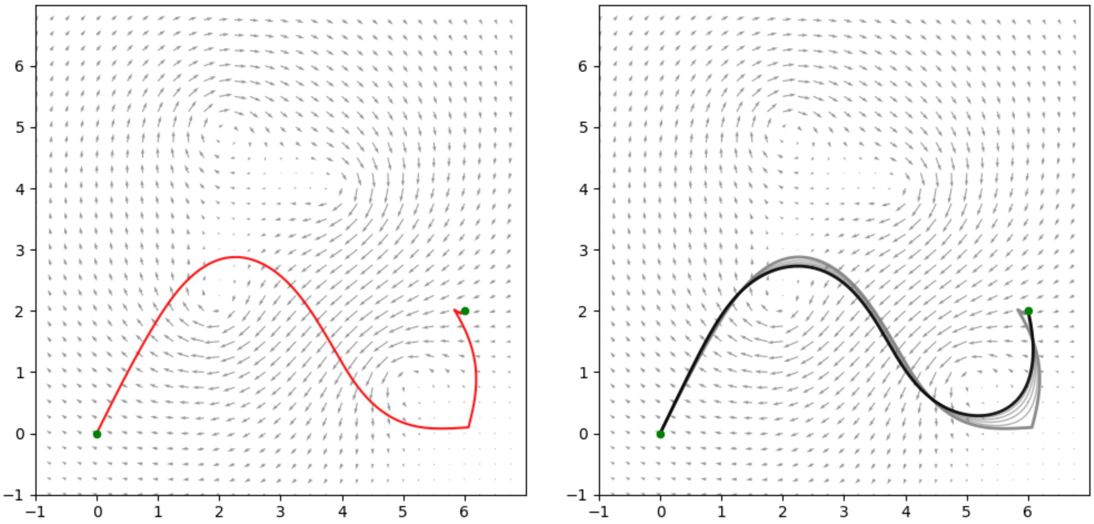
Let us quickly review the Newton-Jacobi iterative procedure for solving non-linear equations. Consider an equation of the form $0 = f(x)$ where $f(x)$ is a differentiable function of 1 variable. Newton's method proposes that we pick an approximate solution $x = x_1^{(0)}$ and then solve the linearized system

$$f(x_1^{(0)}) + f'(x_1^{(0)})(x_1 - x_1^{(0)}) = 0$$

to obtain an x_1. If $x_1^{(0)}$ is sufficiently close to a root of $f(x) = 0$, one can show that $|f(x_1)| < |f(x_1^{(0)})|$ and we can iterate to produce a sequence $x_1^{(0)}, x_1, x_2, \ldots$ by solving, at each stage the linearized system

$$f(x_1(i)) + f'(x_1(i))(x_{i+1} - x_1(i)) = 0.$$

The Newton-Jacobi method generalizes Newton's method to the case of an $n\times n$ system of non-linear equations $F(q) = 0$ where q is a point in n-dimensional space and F is a



(a) (b)

Figure 4.5: (a) Optimized route obtained by alternating the first two steps of Hybrid Search method. The segments are locally optimal (thanks to RK4) but are joined by sharp turns. (b) The whole route is then smoothed with Ferrero-de Diego-Almagro Algorithm method for 10 000 iterations.

transformation of n-dimensional space; i.e., $F(q) = (F_1(q), \dots, F_n(q))$ is an n-vector of functions. As above, we begin with an initial guess q_0 and then construct a sequence of approximate solutions by solving the linearized equations

$$F(q_i) + DF(q_i)(q_{i+1} - q_i) = 0$$

for q_{i+1}. Under suitable assumptions, one can show that the sequence $q_0, q_1, q_2, \dots$ converges to a zero of F.

The discrete Euler-Lagrange equations (4.36) are a non-linear system of equations of $n \times (N-1)$ equations. The key idea introduced in Ferraro, de Diego and Almagro [144] is to apply the Newton-Jacobi method iteratively to primitive 3-point trajectories, i.e. segments consisting of q_{k-1}, q_k, q_{k+1}. For each such trajectory we freeze q_{k-1}, q_{k+1} and seek for the optimal placement of q_k. This amounts to a solution of the equation

$$D_2L_d(q_{k-1}, \bar{q}_k) + D_1L_d(\bar{q}_k, q_{k+1}) = 0$$

for an unknown $\bar{q}_k$. We now apply the Newton-Jacobi method by taking

$$F(q) = D_2L_d(q_{k-1}, q) + D_1L_d(q, q_{k+1})$$

and apply one iteration of the method to solve the linearized system

$$F(q_k) + DF(q_k)(q_k^* - q_k) = 0$$

for the unknown q_k^*. Fully written, the system for q_k^* is then

$$\begin{aligned} &D_2L_d(q_{k-1}, q_k) + D_1L_d(q_k, q_{k+1}) + \\ &\qquad + \left(D_{22}(q_{k-1}, q_k) + D_11L_d(q_k, q_{k+1})\right)(q_k^* - q_k) = 0 \end{aligned}$$

We now apply the same 1-step iteration to all the primitive trajectories

$$(q_{k-1}, q_k, q_{k+1}),\ k = 1, \ldots, N-1$$

to obtain a new trajectory $q^* = (q_k^*)_{k=0}^N$ with $q_0^* = q_0$ and $q_N^* = q_N$. If the initial trajectory $q^{(0)}$ is well chosen, then the iterated sequence of trajectories $q^{(i)}, i = 0, 1, \ldots$ where $q^{(i+1)} = q^{(i)*}$ converges to a solution of the discretized Euler-Lagrange equations (4.36).

We will apply the FDA algorithm to the Euclidean Zermelo problem after suitably transformation (4.18) to a non-constrained optimization problem. It is possible to extend the FDA methodology to spherical backgrounds and to constrained optimization, but we do not pursue these directions in the present study.

We begin by combining (4.19) into the single constraint

$$(\dot{x}_1 - w_1\dot{t})^2 + (\dot{x}_2 - w_2\dot{t})^2 = V^2\dot{t}^2. \tag{4.15}$$

Setting

$$X = \sqrt{\dot{x}_1^2 + \dot{x}_2^2}$$
$$W = \sqrt{w_1^2 + w_2^2}$$

we rewrite (4.15) as the following quadratic equation in $\dot{t}$:

$$(V^2 - W^2)\dot{t}^2 + 2XW\cos\beta - X^2 = 0,$$

where β is the angle between $\dot{x}$ and w. The solution gives us the following unconstrained Lagrangian:

$$\hat{L} = \dot{t} = \frac{X}{V^2 - W^2}\left(-W\cos\beta + \sqrt{V^2 - W^2\sin^2\beta}\right)$$

As given, the above $\hat{L}$ is not a regular Lagrangian, and the corresponding $\hat{L}_d$ will not give a convergent FDA algorithm. This difficulty can be remedied by observing that $\hat{L}^2$ **is** regular, and so we take $\hat{L}_d^2$ as the discrete Lagrangian for our implementation of the FDA algorithm.

Figure 4.5b shows the results of FDA after 10 000 iterations, applied to the route generated at the end of the exploration and refinement loop.

4.4.2. Graph Optimization

Graph optimization is a powerful mathematical technique that is widely used in the field of weather routing [75]. This type of algorithms represent the ocean as a graph and find the path that minimizes a specific objective function. The objective function may be designed to minimize the travel time, fuel consumption, or other operational costs, depending on the requirements of the application. One advantage of graph search algorithms is their ease to implement constraints, such as obstacle avoidance, just by disallowing some nodes in the graph.

To implement graph optimization, we first have to discretize the whole world map using a grid. Each node of the grid correspond to a coordinate in latitude-longitude. The grid avoids points on land and waters where the vessel should not be - due to piracy or no sail zones. Following the resolution seen on Table 4.1, the distance between points is 0.08° (roughly 9 km, depending on the latitude). This gives a grid of size 4500×2250, containing over 10 million nodes. The grid is transformed into an undirected unweighted graph, where each node is a coordinate and their edges connect adjacent zones.

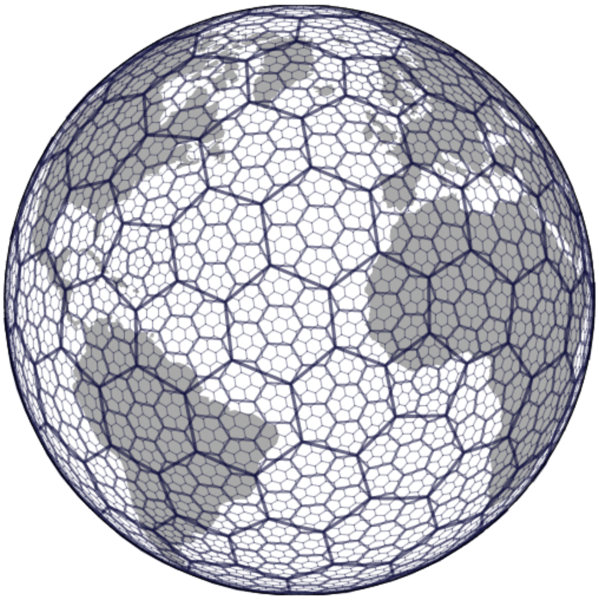


Figure 4.6: Partition of the Earth into hexagonal grids of different sizes, using H3. Image from Uber Technologies [169].

Using the square grid described above has its shortcomings. For starters, the spherical shape of the Earth forces a big distortion in order to fit the square grid. Our solution was to use instead an hexagonal grid provided by the library H3 [169]. As shown in Figure 4.6, hexagons are very accurate when covering the round shape of Earth. In addition, each hexagon has six neighbours, giving more possible directions for the route to follow, compared to a squared grid that only has four neighbours and thus four orthogonal directions. This is important as real vessels take smooth turns: we want to give as many directions possible to each node. In fact, six angles still forces the algorithm to take sharp turns, as the minimum course correction would be 60°. To fix this issue, we can connect each node to its N-order neighbours, effectively giving $6 \cdot (1 + \sum_{k=0}^{N-1} k)$ possible directions, as shown in Figure 4.7.

We also have to take into account that the data depends on time, due to the variable oceanography. This problem can be addressed by adding an extra dimension to the graph. We end with a 3-dimensional graph, having coordinates across latitude, longitude and

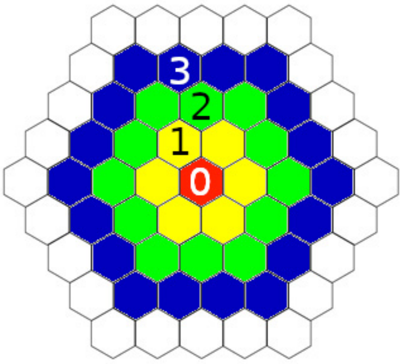


Figure 4.7: Hexagonal graph. Taking the red node as reference, each other is coloured according to the number of jumps required to reach them in a graph of 1-order. Meanwhile, a graph with 3-order connections allows the algorithm to jump from the red node directly to any green or blue node, giving 36 possible nodes and 24 different directions.

time. Graph timestamps are evenly spread following a set time delta. This means that time is also discrete, and we risk losing meteorological events that happen in lapses shorter that our time delta.

Having built the graph, our goal is to find the minimal path between two nodes $(n_{\text{start}}, n_{\text{end}})$. We define the minimal path as the one with fewer fuel consumption, following the equation (4.1). The most popular method used in graph optimization problems is Dijkstra [170]. This algorithm computes the minimal path between each pair of nodes, so it must explore all of them. Its order of complexity, using an optimal data structure, is:

$$O\left(|E| \log |N|\right)$$

where E is the set of edges and N is the set of nodes in the graph. The problem we face in this case is that we cannot explore the whole graph due to its size (over 10 million nodes). The alternative is A*[171]. Contrary to Dijkstra, A* does not need to explore the whole graph to find the minimal path between two nodes. The cost function of A* for node n is:

$$f(n) = g(n) + h(n) \tag{4.16}$$

where $g(n)$ is the cost of the path from n_{start} to n, and $h(n)$ is the heuristic cost of the path from n to n_{end}. This algorithm takes into account both the path that has been explored and an approximation of what needs to be traversed. Thus, A* gives an optimal solution as long as the heuristic is admissible. The heuristic should not overestimate the real cost from n to n_{end}. On the other hand, better heuristics will reduce the need of the algorithm to explore nodes. For this reason, the complexity of A* depends on the chosen heuristic. In the worst case scenario, a bad heuristic will force A* algorithm to explore every node of the graph, just like Dijkstra does.

There exists several variations of A* that can help the algorithm to converge faster. One of then is Weighted A*[172], whose cost function is:

$$f(n) = g(n) + w * h(n) \tag{4.17}$$

where w is a weight that multiplies the heuristic component. This affects how many nodes are explored by the algorithm. For instance $w = 0$ cancels the heuristic component, making A* behave just like Dijkstra. On the other hand, if we assign a very high value to w, the path from n_{start} to n will not be taken into account, and thus the algorithm will turn into a Greedy Best First Search [173]. Higher values of w will speed up the algorithm but they do not ensure the optimal result, even when using admissible heuristics.

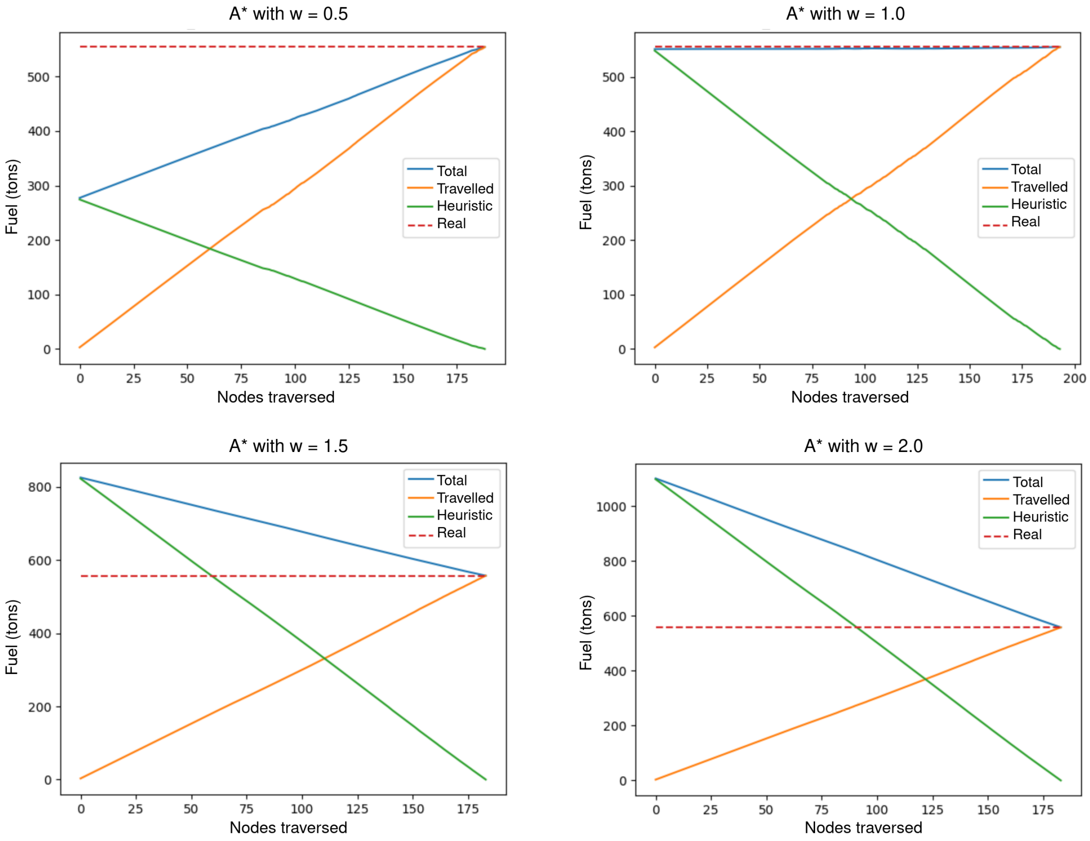


Figure 4.8: Effect of the weight w on the A* cost function. Blue line is the total cost function $f(n)$, orange is the cost from the origin to the current node $g(n)$, and green is the expected cost from the current node to the goal i.e. the heuristic $h(n)$. For reference, red line is the real cost of the journey.

Figure 4.8 shows, for different values of w, how the cost function f and its components g, h vary while the route progresses. For $w < 1$, cost function values fall below the real consumption, because the heuristic is being underused. For $w = 1$, the cost function is very close to the real consumption thanks to the heuristic. For $w > 1$, the heuristic is overused and makes the cost function surpass the real consumption, causing strange behaviour in the algorithm. This study on the effect of w was done using a vessel with average speed and following a simple journey: Charleston (USA) to Azores archipelago, during summer (described in section 4.3.2). Having done this study, we conclude that the best weight w for our A* must be close to 1. The chosen heuristic with $w = 1$ is very accurate for higher vessel speeds, as the ocean currents will have lesser impact on its trajectory. We want to point out that when land is present between start and goal the heuristic will underestimate the cost, as it assumes a straight path.

4.4.3. Genetic Algorithm

The Darwinian idea of evolution inspired the Genetic Algorithm (GA), which simulates the survival of fitter individual and their genes [174], [175]. The GA algorithm is a population-based algorithm. In the context of weather routing, any route represents an individual (also named chromosome or solution), and its genes may be, for instance, the waypoints of that route, or the heading of vessel at each timestamp. GA uses a fitness (objective) function to assess the suitability of each member in the population. For our case study, this fitness function computes the fuel consumption of the route. To improve unsatisfactory solutions, the best chromosomes are picked at random using a selection operator. This mechanism takes fitness into account, so that the best solutions are more likely to be selected. However, the possibility of selecting poor routes also enhances the likelihood of avoiding local optima: if good solutions become stuck in a local solution, they can be pulled out using other solutions.

The process of keeping the best route in each generation and applying them to enhance subsequent guesses is what makes GA reliable and able to estimate the global optimum for a particular journey. As a result, the entire population improves generation after generation. The crossover between routes results in exploiting the “area” between the given two parents. Mutation helps this method as well. This operator modifies the genes on the chromosomes at random, preserving the variety of the population and increasing GA’s exploratory behaviour.

Applying GA to weather routing requires defining what a chromosome and its genes are. As we mentioned earlier, this study assumes a chromosome is a route, and each gene is a waypoint $\boldsymbol{x}(t)$. As such, an individual is defined as a list of length L containing $\boldsymbol{x}(t_i), i \in [0, L]$, being $\boldsymbol{x}(0) = \boldsymbol{x_A}$ the coordinates from where the vessel departures, $\boldsymbol{x}(t_L) = \boldsymbol{x_B}$ the goal. All the other coordinates will be chosen by the GA, and times t_i will be computed based on these coordinates, the vessel speed and the weather conditions.

Allowing the GA to generate random coordinates at any point in the ocean is extremely inefficient. Its guesses would cover very long distances, as the waypoints will appear very sparse, and the convergence will take very long time to happen, while also risking getting stuck on more local minima. For these reasons, limiting the GA search space is mandatory to accelerate its convergence into optimal results. However, we do not want to set very strict limitations, as these would kill the exploratory nature of the GA, which is one of its advantages.

The process we are going to follow is the following:

1. First, we compute the geodesic between the start ($\boldsymbol{x_A}$) and end ($\boldsymbol{x_B}$) coordinates, namely $\vec{g}$. The geodesic is the line of minimum distance. Thus, optimal solutions are often found around the geodesic, specially at higher vessel speeds.
2. Next, we define L equidistant waypoints across the geodesic,

   $$q_j : q_j \in \vec{g}, \; \forall j \in [0, L].$$

   By definition, $q_0 = \boldsymbol{x_A}$ and $q_L = \boldsymbol{x_B}$.
3. For points q_2 to q_{L-1}, we generate lines $\vec{l}_j$ crossing them, perpendicular to the geodesic:

   $$\vec{l}_j : q_j \in \vec{l}_j, \; \vec{l}_j \perp \vec{g}, \; \forall j \in [1, L-1].$$

   The length of these lines is $w \cdot D(\boldsymbol{x_A}, \boldsymbol{x_B})$, where w is called the width of the search space around the geodesic. The middle of these lines intersects the geodesic. Line $\vec{l}_j$ will contain the possible values of gene j for the GA. Genes 0 and L are fixed, as they are the known start and end coordinates.
4. We define sets of equidistant points $s_{j,k}$ across these new lines, having a separation of Δs between consecutive values,

   $$s_{j,k} \in \vec{l}_j, \; \forall k, \; \forall j \in [1, L-1].$$

   So that $s_{j,k}$ are the possible values of the gene j.

With this, a grid of possible positions is defined, ensuring that all solutions follow a “reasonable” path to the goal - in the sense that they will always travel towards the goal. Figure 4.9 illustrates this grid and shows an example of a route generated with it.

This definition of genes also ensure that any crossing between individuals will generate another reasonable route: any value of gene j is always a step towards the goal from gene $j - 1$. Additionally, by removing the waypoints on land in the grid, we minimize the chance of generated routes to travel across mainland.

During each generation, two parents are chosen out of all the population, using tournament selection. This implies splitting the routes into two groups, then picking the individual with the least fuel consumption within each group. Having chosen the parents, they generate two children. With a probability p_1, the parents recombine by performing a two-point crossover between them. This implies slicing each parent chromosome into three segments of variable size (but equal for both individuals), and exchanging the middle section from one solution with the other. The result will be two children each

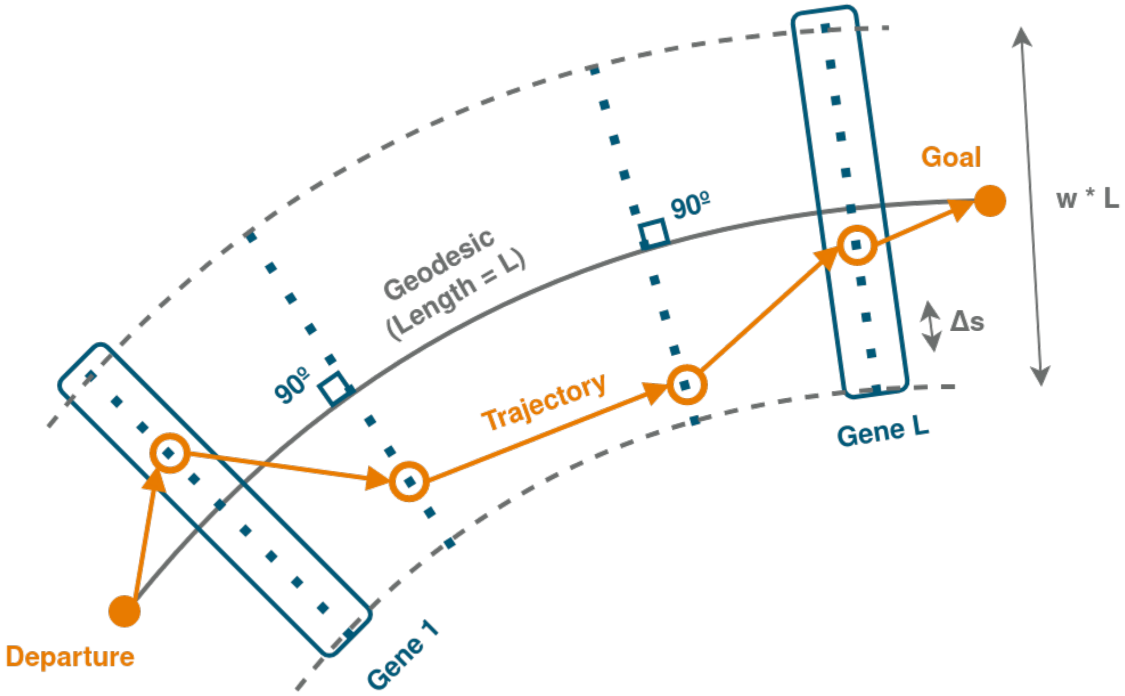


Figure 4.9: Grid defined by the Genetic Algorithm (GA), following the geodesic. Each set of blue points (perpendicular to the geodesic) are the possible values that a gene can take. In orange, we show a possible route that can be generated by this algorithm.

having characteristics of both parents. If the parents do not recombine, the children will just be copies of the parents. Once the children are defined, each of their genes may mutate with a probability p_2, changing into a random value within the allowed (any $s_{j,k}$ for gene j). Finally, the children routes are evaluated: any child route that consumes less than the worst member of the population will enter, replacing the previous worst chromosome.

This process keeps going until a set number of generations have passed. We can also define more fine-tuned stopping criteria, such as waiting for a number of generations to pass without any solution improving over the best one.

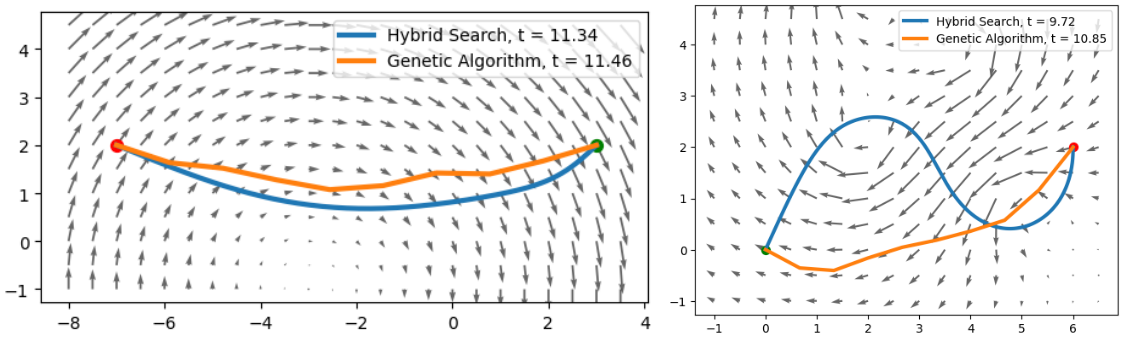
4.5. Results

In this section we run all the algorithms described in section 4.4 (Hybrid Search, A*and the genetic algorithm) on each benchmark mentioned in section 4.3.2, aiming to find the route that is optimal under each criteria, which varies over optimization methods. To have a point of comparison, we also show the time elapsed by the route of minimum distance. For the synthetic vector fields, operating in Euclidean geometry, the minimum distance is the straight line. For the real vector fields, the minimum distance is the geodesic in absence of land, otherwise it is called the circumnavigation route (route of minimum distance avoiding land).

Prototypes of the algorithms were implemented using Python, in some cases using the JAX library [176]. The simulations were run on an Intel Core i7-8700K CPU (3.7 GHz), and an NVIDIA GeForce GTX 1080 GPU (8 GB). The execution times provided throughout this section are given only for illustration purposes and transparency. However, these are strongly dependent on the implementation, which may have been suboptimal.

4.5.1. Synthetic benchmarks

To solve the synthetic vector fields for a vessel of unitary velocity on Euclidean geometry, Hybrid Search (HS) (section 4.4.1) was run using a time step of $\Delta t = 0.01$ and checking the stopping criteria every $\tau = 0.1$. There were twenty one trajectories being tested by the HS, their initial shootings evenly spread across a cone of amplitude $\gamma = \pi$ centred on the direction to the goal. The trajectories would stop if their heading deviated more than $\gamma_d = \pi/2$, or when they got close to the goal (at least $d = 0.1$). Once the HS guessed an optimal route, it would be smoothed by Ferrero-de Diego-Almagro Algorithm (FDA) during 10 000 iterations.



(a) Circular

(b) Four vortices

Figure 4.10: Results on the synthetic vector fields using the Hybrid Search (HS) variational method (blue) and the genetic algorithm (orange), sailing at unitary velocity. Green dot marks the departure and red dot is the goal.

For the genetic algorithm (section 4.4.3), there were $L = 100$ genes, distributed along a width of $w = 0.3$ and a step of $\Delta s = 0.2$. There were 60 chromosomes on the population and they evolved through 500 generations, with a recombination probability of $p_1 = 0.8$ and mutation probability of $p_2 = 0.2$.

The travel times obtained by our methods in the two synthetic benchmarks are shown in Table 4.2. A*search could not be applied as it was designed to work on spherical space, not in the plane. It will be tested later on the real benchmarks. In addition to the table, we show the optimized routes in Figure 4.10.

Vector field	Method	Time	Comp. time (min)
Circular	Min. distance	11.93	-
	Hybrid Search	**11.34**	**1**
	Genetic	11.46	2
Four Vortices	Min. distance	30.44	-
	Hybrid Search	**9.72**	**1**
	Genetic	10.85	3

Table 4.2: Results on synthetic vector fields with unitary velocity, comparing the route of minimum distance with the output from our routing methods. Computation time is provided for illustration purposes.

4.5.2. Real benchmarks

To solve the real vector fields for a vessel of different velocities on spherical geometry (approximating the Earth's radius as 6367.449 km), Hybrid Search (HS) was run using a time step of $\Delta t = 600$ s (10 minutes) and checking the stopping criteria every $\tau = 7200$ s (2 hours). HS aimed to optimize the time of the journey, not taking the fuel consumption into account. There were twenty one trajectories being tested by the HS, their initial shootings evenly spread across a cone of amplitude $\gamma = \pi/3$ (60°) centred on the direction to the goal. The trajectories would stop if their heading deviated more than $\gamma_d = 2\pi/3$ (120°), or when they got close to the goal (at least $d = 10$ km). Once the HS guessed an optimal route, it would be smoothed by Ferrero-de Diego-Almagro Algorithm (FDA) during 2000 iterations. Figure 4.11 shows its result on the journey from Charleston from Azores, in comparison with the route of minimum distance.

For the genetic algorithm (section 4.4.3), there were $L = 100$ genes, distributed along a width of $w = 0.3$ and a step of $\Delta s = 10$ km. There were 60 chromosomes on the population and they evolved through 500 generations, with a recombination probability of $p_1 = 0.8$ and mutation probability of $p_2 = 0.2$. Fuel consumption was taken as the cost function for the genetic algorithms, assuming the dimensions of a generic PCTC vessel[1]: length $L = 297$ m and beam $B = 46$ m. It is relevant to note that the fuel consumed only considered the effect of currents and waves (equations (4.3) and (4.5) respectively). Wind was not taken into account as at the time of this study we lack the required vessel parameters to solve equation (4.7).

Finally, the A*search algorithm used the fuel consumption as cost function, just like the genetic algorithm did (i.e. equations (4.3) and (4.5)). During search, the algorithm was allowed to jump to neighbouring nodes up to 3rd order, as shown in Figure 4.7. Its heuristic was given a weight of $w = 1.05$ (see equation (4.17)), in an attempt to prioritize

---

[1] Flex Aurora, IMO 9857365, `https://www.shipspotting.com/photos/3073441`

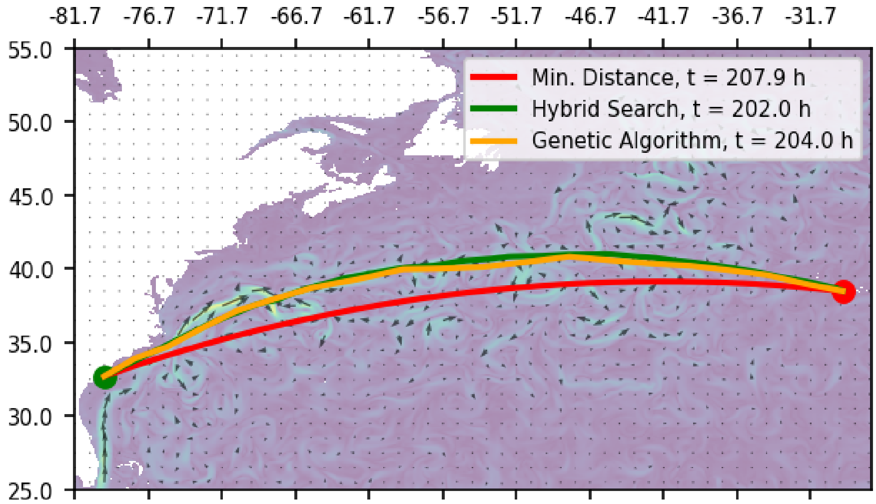


Figure 4.11: Results of the Hybrid Search variational method (yellow) on the journey from Charleston to Azores, sailing at 6 m/s. The genetic algorithm route (not shown) follows the minimum distance (red) closely.

Speed (m/s)	**Method**	**Travel time** (h)	**Distance** (km)	**Fuel** (tons)	**Comp. time** (min)
3	Min. distance	416.4	**4392.5**	97.9	-
	Hybrid Search	**389.2**	4426.4	83.9	9
	$A^\star$search	395.6	4480.4	**73.9**	**4**
	Genetic	393.1	4459.6	78.9	7
6	Min. distance	207.9	**4392.5**	255.4	-
	Hybrid Search	**202.0**	4423.3	238.2	6
	$A^\star$search	203.7	4458.4	**229.1**	**2**
	Genetic	204.0	4456.5	234.7	7
10	Min. distance	124.9	**4392.5**	703.6	-
	Hybrid Search	**123.3**	4418.3	683.5	4
	$A^\star$search	123.7	4436.4	**674.5**	**1**
	Genetic	124.5	4451.5	684.4	7

Table 4.3: Results on the journey Charleston - Azores, departing at the 25th of May of 2022. We compare the route of minimum distance (geodesic) with the output from our routing methods. Computation time is provided for illustration purposes.

routes that head straight to the goal, speeding the computation at the expense of skipping some exploration.

Results of all three algorithms are shown on table 4.3 for Charleston to Azores and table 4.4 for Somalia to Myanmar. Their routes are compared using the time they take to be traversed, the distance they cover, and the fuel consumed assuming. Algorithms were also compared by their computation time.

Speed (m/s)	**Method**	**Travel time** (h)	**Distance** (km)	**Fuel** (tons)	**Comp. time** (min)
3	Min. distance	552.2	**6159.8**	50.3	-
	Hybrid Search	**528.3**	6196.9	46.4	10
	A*search	535.6	6344.6	**44.3**	**3**
	Genetic	529.3	6243.4	44.9	5
6	Min. distance	280.0	**6159.8**	263.2	-
	Hybrid Search	**274.8**	6196.9	256.4	8
	A*search	277.2	6252.2	255.9	**3**
	Genetic	275.3	6193.5	**255.3**	6
10	Min. distance	169.3	**6159.8**	872.3	-
	Hybrid Search	**167.8**	6190.4	862.6	6
	A*search	168.5	6221.4	863.3	**2**
	Genetic	168.0	6183.8	**862.2**	5

Table 4.4: Results on the journey Somalia - Myanmar, departing at the 1st of July of 2022. We compare the route of minimum distance (circumnavigation) with the output from our optimization methods. Computation time is provided for illustration purposes.

4.5.3. Seasonal differences

An intriguing avenue for further investigation involves examining the variation in routes over the course of a year, taking into account the influence of weather changes on algorithmic decision-making. This study will focus on the A*search method presented in section 4.4.2, as it has demonstrated consistently favourable results and boasts considerably greater computational speed than alternative optimization algorithms. The vessel will be assumed to sail at 10 meters per second, as this is an usual design speed.

Our first seasonal study focuses on the Atlantic Ocean, based on the benchmark from section 4.3.2. Charleston serves once again as the departure port, but with the destination relocated from Azores Island to Algeciras. This modification doubles the journey's length, affording greater opportunity to alter vessel course and capitalize on varying weather conditions, particularly the powerful currents of the Gulf and North Atlantic. A total of 52 routes will be simulated, one for each week between September 1, 2021 and August 31, 2022, providing a comprehensive view across the year.

For each simulation, we first computed the consumption of the shortest route during each week of the year, which is plotted in figure 4.12. The average consumption of this geodesic route is 962t of fuel, but significant variations are observed, ranging from -2% to 4%, which shows that the effect of the weather, without modifying the route, already have an impact on fuel consumption. It is noteworthy, for instance, that the distribution of variations is not symmetrical with respect to the mean, i.e. it is not a normal distribution. In fact, there are some points of consumption well above the mean that are due to the

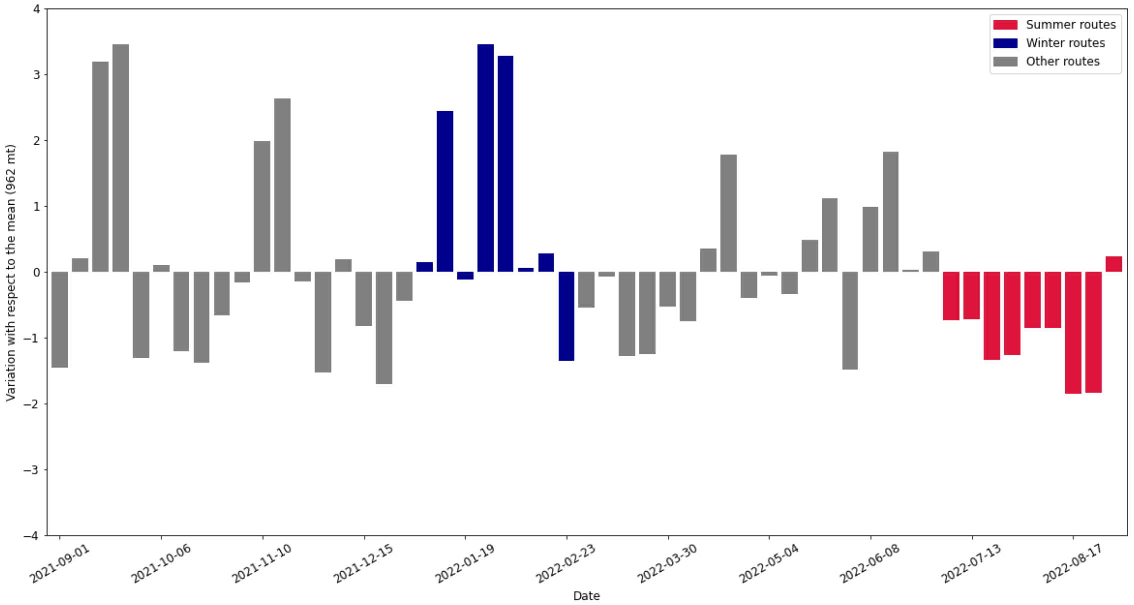


Figure 4.12: Variation of fuel consumption for the shortest route at each time of the year with respect to the average consumption (962 tons of fuel).

presence of specific meteorological events, such as storms that produce areas of high waves. For the seasonal comparison we have highlighted 10 simulations in successive weeks during winter (blue colour) and in summer (red colour), observing a lower fuel consumption in summer due to better sailing conditions in the northern hemisphere.

Figure 4.13: Trajectories generated with $A^{\star}$ search method during summer (July and August, in red) and winter time (January and February, in blue), following the journey from Charleston (USA) to Algeciras (Spain).

Next, $A^{\star}$search algorithm was applied to compute the optimal route for each week. Figure 4.13 shows some of these optimized journeys, grouped into summer and winter times. Then, figure 4.14 demonstrates that, over the course of 52 weeks, the optimized route consistently results in cost savings when compared to the shortest distance route, despite entailing a longer travel distance.

Building on our previous research conducted on the Atlantic Ocean, we extend our analysis to the Indian Ocean. The Indian Ocean presents an intriguing feature in the form of the Somalia current, which changes morphology over the year [177]. In this study,

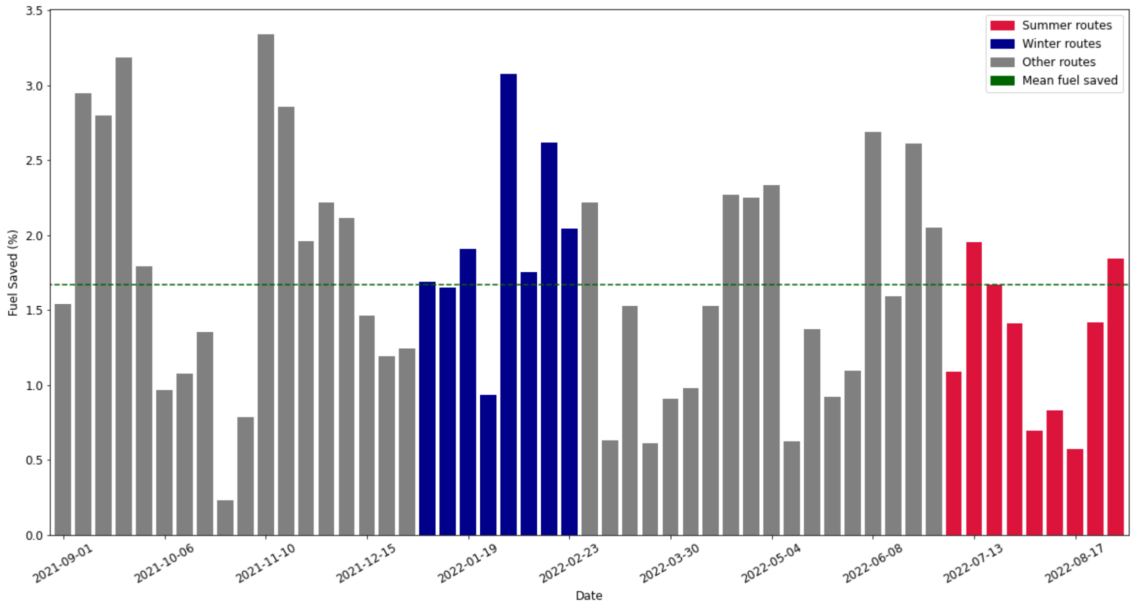


Figure 4.14: Fuel saved by $A^\star$ generated routes at each time of the year, compared to the shortest route possible.

we follow the benchmark from section 4.3.2, selecting the Somali coast in East Africa as the departure point and Myanmar in Asia as the destination port. We perform multiple simulations covering dates between January and July of 2022. Our baseline will be the circumnavigation route, i.e., the geodesic route that is closest to the ship's course without running aground.

Season	**Avg. fuel consumption (tons)**	**Avg. fuel saved**	**Avg. time reduced**
Winter	982.79	0.6%	1.3%
Summer	947.36	1.9%	0.9%

Table 4.5: Fuel consumed by $A^\star$ generated routes on the Indian Ocean, compared to circumnavigating.

Table 4.5 presents the results obtained from the comparison between the optimized routes and the base route, and distinguishes between the winter and summer periods. Several notable observations emerged from the analysis, including the following findings: During the winter season, the navigational direction of a boat is often forced to run counter to the direction of the current, resulting in an elevated average fuel consumption rate of 983t when compared to summer. To mitigate this issue, our optimization algorithm was utilized to identify alternative routes. Our findings indicate that the employment of optimized routes yielded an average fuel cost reduction of 0.6%. During the summer season, the prevailing weather conditions are conducive to optimizing transportation routes, resulting in lower average fuel consumption (947t). Additionally, our analysis indicates that the optimizer algorithm identifies greater opportunities for improvement during this period, as the optimized routes yield an average fuel savings of 1.9% (with a maximum savings of 2.5% observed on a specific day).

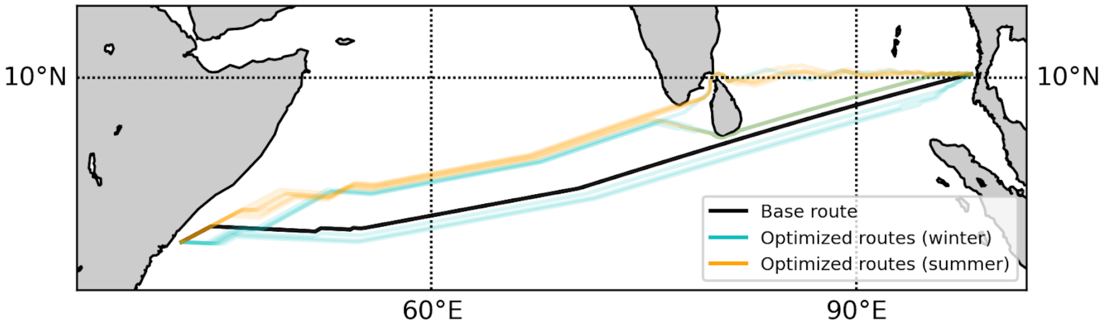


Figure 4.15: A* generated routes on the Indian Ocean, from Somalia (west) to Myanmar (east). Routes are coloured by season: summer in orange, winter in blue. The circumnavigation, used as base route, is shown in black.

To provide a visual representation of the routes, we present a projection of the region in figure 4.15 for illustrative purposes. The reference path is displayed in black, while the various routes discovered by the optimizer discussed in section 4.4.2 are represented in orange and blue, corresponding to each day of the respective seasonal periods.

4.6. Discussion

This study on weather routing explores the effectiveness of three optimization algorithms: a variational method named Hybrid Search (HS) (section 4.4.1), a genetic algorithm (section 4.4.3) and an A*graph search (section 4.4.2). These algorithms were initially tested on synthetic benchmarks, and subsequently, on two real case examples. Before starting the discussion, it is important to note that the study of these algorithms is still in its early stages, and for this thesis, only a small number of routes were compared. The results presented here should be viewed as an illustration of how these computational methods can improve standard shipping routes. It is crucial to note that these results do not provide conclusive proof that any algorithm is superior to the others. In fact, the findings suggest that all algorithms perform similarly well and that further development of any of them can provide effective weather routing.

The results obtained from the synthetic benchmarks show that both the HS method and the genetic algorithm outperform the minimum distance route in terms of travel time. The HS method exhibits slight better results over the genetic algorithm, and it is also faster in computational time. This was to be expected, as variational methods are meant for continuous vector fields. Notably, the HS method performs exceptionally well in the four vortices vector field, where the velocity of the currents is similar to that of the vessel, and hence the currents have a significant impact. Figure 4.10b shows that the route found by HS is coincidentally one of the fastest routes discovered by the benchmark's original designers [144], validating the HS method's ability to provide optimal solutions for synthetic benchmarks. Conversely, the circular vector field has lower current speeds, resulting in optimized routes that are not substantially different from the minimum distance route. It is worth noting that the $A^{\star}$graph search algorithm was not used for the synthetic benchmarks since it is designed to work on a spherical plane, while the benchmarks were Euclidean. Therefore, it was not feasible to apply the graph search algorithm in this scenario.

In terms of the real benchmarks, we conducted an analysis of two examples situated in the Atlantic and Indian oceans. Three distinct sailing speeds were tested, including one at the design speed (ten meters per second) and two lower speeds (three and six meters per second). The findings in Tables 4.3 and 4.4 show that all optimization methods resulted in fuel savings, with the $A^{\star}$search algorithm demonstrating the most significant improvements. The HS method, on the other hand, excelled in identifying the shortest time routes. This different in performance can be attributed to the distinct operating principles of the two methods: while HS prioritizes time optimal trajectories by following favourable currents, $A^{\star}$search focuses on minimizing the cost function (fuel consumption) and considers the impact of waves. Meanwhile, the genetic algorithm found a balance between travel time and fuel consumption, proving to be better at faster sailing speeds.

Our findings from real-life benchmarks provide insight into the impact of Harvald's law, shown in equation (4.3), on vessel performance. Harvald's law suggests that engine power increases cubically with speed, leading to a quadratic increase in fuel consumption with vessel velocity (one order less as increasing speed reduces travel time). Tables 4.3 and 4.4 demonstrate this effect, although wave effects also come into play. It is evident that reducing vessel velocity significantly decreases fuel consumption, specially when wave height is low. Such is the case in the Indian Ocean, where the mean wave height is one meter, compared to the Atlantic Ocean, where it is over three meter [178]. In any case, these results highlight the importance of adopting **slow steaming** practices, which will be commented on the conclusion.

Given the success of our weather routing algorithms at saving fuel, and the computational efficiency of the $A^{\star}$graph search method, we conducted a follow-up study to investigate how seasonal weather changes impact the optimal route. Specifically, we wanted to determine how variations in ocean currents and wave heights affected the route recommended by the algorithm, when sailing at the design speed of ten meters per second (twelve knots). By understanding these seasonal weather patterns, we hoped to identify opportunities to further optimize shipping operations and reduce fuel consumption.

The results of our seasonal study on the Atlantic ocean demonstrate that weather routing algorithms can lead to significant cost and environmental savings in the shipping industry. The fuel savings achievable through optimized routes can reach up to 3.3%, with an average savings of 1.7% per route, corresponding to a reduction of approximately 16t of fuel consumption per voyage. These savings have substantial economic and environmental impacts, with potential fuel savings amounting to €15 000 for a typical one-week transatlantic route, and the avoidance of approximately 50t of CO_2 emissions. This underscores the importance of implementing route optimization as a viable means of reducing costs and mitigating the environmental impact of shipping operations.

Further analysis of our results on the Atlantic reveals that the most substantial fuel savings are achieved in routes where the geodesic route's energy consumption is higher. This observation can be attributed to the presence of adverse navigation conditions and inclement weather, which the optimized routes successfully circumvent to a greater extent. These findings highlight the importance of accounting for weather and ocean conditions when optimizing shipping routes, as they can have a significant impact on fuel consumption and emissions. Additionally, our study shows that optimal routes in the summer tend to depart from a more northerly position to take advantage of the Gulf current before turning south to approach the geodesic route. Conversely, in the winter months, the optimal routes tend to travel further north but benefit from better navigation conditions. These observations suggest that implementing weather routing algorithms can lead to more efficient and cost-effective shipping operations throughout the year, and especially when combined with slow steaming practices, which are becoming more common in the industry.

The findings of the second seasonal study in the Indian ocean highlight the benefits of weather routing optimization beyond fuel efficiency. The optimized routes not only yielded fuel savings but also resulted in shorter travel durations when compared to the reference path. This result emphasizes the need for further investigation into the multiple benefits of optimizing travel routes. The study also showed that the optimized routes varied significantly depending on the seasonal period, indicating the efficacy of the weather routing optimizer in leveraging external information. The statistically significant above-average savings observed in both scenarios further support the effectiveness of the weather routing optimization technique in reducing costs and mitigating the environmental impact of shipping operations. These findings demonstrate the importance of considering weather patterns and seasonal variability in the design and optimization of shipping routes to improve their efficiency and sustainability.

In conclusion, our study on weather routing algorithms has demonstrated that optimization methods can significantly reduce fuel consumption and emissions in the shipping industry. This study also highlights the importance of accounting for weather patterns and ocean conditions when optimizing shipping routes to achieve more cost-effective and sustainable operations. These findings suggest that the implementation of weather routing algorithms can lead to substantial economic and environmental benefits, including fuel savings of up to 3.3% and the reduction of CO_2 emissions. Overall, this chapter underscores the potential of weather routing optimization as a viable means of improving the efficiency and sustainability of shipping operations. While acknowledging the existence of further work to be accomplished, the elaboration on this matter is deferred to Chapter 5.

APPENDICES

4.A. Derivation of Zermelo's equations

4.A.1. Zermelo's Navigation Problem on the plane

We are dealing here with a constrained optimization problem whose Lagrangian function has the form

$$L = \dot{t} + \lambda_1(\dot{x}_1 - (V\cos\alpha + w_1)\dot{t}) + \lambda_2(\dot{x}_2 - (V\sin\alpha + w_2)\dot{t}). \tag{4.18}$$

The goal is to find trajectories $x(s)$, $\dot{x}(s) = x'(s)$, $t(s)$, $\dot{t}(s) = t'(s) > 0$, $\alpha(s)$ with fixed end-points that minimize $t(s_1) - t(s_0) = \int_{s_0}^{s_1} L ds$, and obey constraints

$$\begin{aligned} \dot{x}_1 &= (V\cos\alpha + w_1)\dot{t} \\ \dot{x}_2 &= (V\sin\alpha + w_2)\dot{t} \end{aligned} \tag{4.19}$$

The quantities λ_1, λ_2 are known as Lagrange multipliers. As we now show, their form is determined by the Euler-Lagrange equations associated with the above Lagrangian, namely

$$\frac{dL_{\dot{t}}}{ds} = 0 \tag{4.20}$$

$$L_{x_i} - \frac{dL_{\dot{x}_i}}{ds} = 0 \quad i = 1, 2 \tag{4.21}$$

$$L_\alpha = 0, \tag{4.22}$$

Equation (4.20) gives

$$\frac{d}{ds}\left(\lambda_1(V\cos\alpha + w_1) + \lambda_2(V\sin\alpha + w_2)\right) = 0$$

which implies that

$$\lambda_1(V\cos\alpha + w_1) + \lambda_2(V\sin\alpha + w_2) = C \tag{4.23}$$

where $C \neq 0$ is a constant. Equation (4.22) gives

$$\lambda_1 \sin\alpha - \lambda_2 \cos\alpha = 0. \tag{4.24}$$

Together, (4.23) (4.24) determine the form of the Lagrange multipliers, namely:

$$\lambda_1 = \frac{C\cos\alpha}{V + w_1\cos\alpha + w_2\sin\alpha} \tag{4.25}$$

$$\lambda_2 = \frac{C\sin\alpha}{V + w_1\cos\alpha + w_2\sin\alpha} \tag{4.26}$$

Going forward, we re-parametrize all curves with respect to time t so that

$$\frac{d}{dt} = \frac{1}{\dot{t}}\frac{d}{ds}.$$

E-L equations (4.21) give the dynamics of the Lagrange multipliers, namely

$$\frac{d\lambda_1}{dt} = -\lambda_1 w_{1,1} - \lambda_2 w_{2,1} \tag{4.27}$$

$$\frac{d\lambda_2}{dt} = -\lambda_1 w_{1,2} - \lambda_2 w_{2,2} \tag{4.28}$$

Rewriting (4.24) as

$$\tan\alpha = \frac{\lambda_2}{\lambda_1},$$

and taking derivatives, gives

$$\begin{aligned}
\sec^2(\alpha)\frac{d\alpha}{dt} &= \frac{d}{dt}\left(\frac{\lambda_2}{\lambda_1}\right)\\
\left(\frac{\lambda_1^2+\lambda_2^2}{\lambda_1^2}\right)\frac{d\alpha}{dt} &= \frac{1}{\lambda_1^2}\left(-\lambda_2\frac{d\lambda_1}{dt} + \lambda_1\frac{d\lambda_2}{dt}\right)\\
\frac{d\alpha}{dt} &= \frac{\lambda_2^2 w_{2,1} + \lambda_1\lambda_2(w_{1,1}-w_{2,2}) - \lambda_1^2 w_{1,2}}{\lambda_1^2+\lambda_2^2}\\
\frac{d\alpha}{dt} &= \sin^2(\alpha)\, w_{2,1} + \sin(\alpha)\cos(\alpha)\,(w_{1,1}-w_{2,2}) - \cos^2(\alpha)\, w_{1,2}
\end{aligned} \tag{4.29}$$

4.A.2. Zermelo's Navigation Problem on the sphere

The modified Lagrangian takes the form

$$\begin{aligned}
L = \dot{t} &+ \lambda_1\left(\dot{\theta} - K^{-1}\sec(\kappa\phi)\,(V\cos(\kappa\alpha) + w_1)\,\dot{t}\right)\\
&+ \lambda_2\left(\dot{\phi} - K^{-1}\,(V\sin(\kappa\alpha) + w_2)\,\dot{t}\right)
\end{aligned}$$

The E-L equations (4.21) now read

$$K\frac{d\lambda_1}{dt} = -\sec(\kappa\phi)\lambda_1 w_{1,1} - \lambda_2 w_{2,1} \tag{4.30}$$

$$K\frac{d\lambda_2}{dt} = -\lambda_1\kappa\sec(\kappa\phi)\tan(\kappa\phi)(V\cos(\kappa\alpha)+w_1) - \lambda_1\sec(\kappa\phi)w_{1,2} - \lambda_2 w_{2,2} \tag{4.31}$$

In the current setting (4.22) gives

$$\tan(\kappa\alpha) = \frac{\lambda_2}{\lambda_1}\cos(\kappa\phi)$$

Taking d/dt yields

$$\begin{aligned}
\kappa \sec^2(\kappa\alpha)\frac{d\alpha}{dt} &= \frac{\cos(\kappa\phi)}{\lambda_1^2}\left(-\lambda_2\frac{d\lambda_1}{dt} + \lambda_1\frac{d\lambda_2}{dt}\right) \\
&\quad - \frac{\kappa}{K}\tan(\kappa\alpha)\tan(\kappa\phi)(V\sin(\kappa\alpha) + w_2) \\
\kappa K \sec^2(\kappa\alpha)\frac{d\alpha}{dt} &= \frac{\lambda_2}{\lambda_1}w_{1,1} + \frac{\lambda_2^2}{\lambda_1^2}\cos(\kappa\phi)w_{2,1} - w_{1,2} - \frac{\lambda_2}{\lambda_1}\cos(\kappa\phi)w_{2,2} \\
&\quad - \kappa\tan(\kappa\phi)(V\cos(\kappa\alpha) + w_1) \\
&\qquad - \kappa\tan(\kappa\alpha)\tan(\kappa\phi)(V\sin(\kappa\alpha) + w_2) \\
\kappa K \sec^2(\kappa\alpha)\frac{d\alpha}{dt} &= \sec(\kappa\phi)\tan(\kappa\alpha)w_{1,1} + \sec(\kappa\phi)\tan^2(\kappa\alpha)w_{2,1} - w_{1,2} \\
&\quad - \tan(\kappa\alpha)w_{2,2} - \tan(\kappa\phi)(V\cos(\kappa\alpha) + w_1) \\
&\qquad - \kappa\tan(\kappa\alpha)\tan(\kappa\phi)(V\sin(\kappa\alpha) + w_2) \\
\kappa K\frac{d\alpha}{dt} &= \begin{bmatrix}\cos(\kappa\alpha) & \sin(\kappa\alpha)\end{bmatrix}\begin{bmatrix}\sec(\kappa\phi)w_{1,1} & w_{1,2} \\ \sec(\kappa\phi)w_{2,1} & w_{2,2}\end{bmatrix}\begin{bmatrix}\sin(\kappa\alpha) \\ -\cos(\kappa\alpha)\end{bmatrix} \\
&\quad - \cos(\kappa\alpha)\tan(\kappa\phi)(V + \cos(\kappa\alpha)w_1 + \sin(\kappa\alpha)w_2) \qquad (4.32)
\end{aligned}$$

4.B. Euler-Lagrange equations

4.B.1. Continuous Euler-Lagrange equations

Define an action functional along a curve $q(t)$ in n-dimensional space with fixed end points as follows,

$$J(q(t)) = \int_a^b L(t, q(t), \dot{q}(t))dt, \quad q(a) = \alpha, \quad q(b) = \beta. \qquad (4.33)$$

The function $L(t, q(t), \dot{q}(t))$ is called the Lagrangian of the optimization problem. The classical problem in the Calculus of Variations is to to minimize J by subjecting $q(t)$ to suitable constraints.

A necessary condition for minimization is that the variation δJ vanishes for all possible variations of the trajectory $\delta q = \epsilon\phi$, where $\phi(t)$ vanishes at the endpoints, and ϵ is the variational parameter. From the functional (4.33), define

$$h(\epsilon) = J(q + \epsilon\phi) = \int_a^b L(t, q(t) + \epsilon\phi(t), \dot{q}(t) + \epsilon\dot{\phi}(t))dt.$$

Now differentiate and use the smoothness of L to interchange the derivative and the integral to get

$$\begin{aligned} h'(\epsilon) = \frac{d}{d\epsilon}J(q+\epsilon\phi) &= \int_a^b \frac{d}{d\epsilon}L\Big(t, q(t)+\epsilon\phi(t), \dot{q}(t)+\epsilon\dot{\phi}(t)\Big)\,dt \\ &= \int_a^b \phi(t)\bigg[\frac{\partial L}{\partial q}\Big(t, q(t)+\epsilon\phi(t), \dot{q}(t)+\epsilon\dot{\phi}(t)\Big) \\ &\qquad + \dot{\phi}(t)\frac{\partial L}{\partial \dot{q}}\Big(t, q(t)+\epsilon\phi(t), \dot{q}(t)+\epsilon\dot{\phi}(t)\Big)\bigg]dt. \end{aligned}$$

Now setting $\epsilon = 0$ and using our definition of the variational derivative yields

$$\delta J(q)(\phi) = \int_a^b \left[\phi(t)\frac{\partial L}{\partial q}(t,q,\dot{q}) + \dot{\phi}(t)\frac{\partial L}{\partial \dot{q}}(t,q,\dot{q})\right]dt. \tag{4.34}$$

This functional is known as the *first variation* of J. In order to obtain an explicit formula for δJ, we need the integral on the right side of the above equation to be linear in $\phi(t)$. We can accomplish this via integration by parts.

$$\int_a^b \dot{\phi}(t)\frac{\partial L}{\partial \dot{q}}(t,q,\dot{q})dt = \left[\phi(t)\frac{\partial L}{\partial \dot{q}}(t,q(t),\dot{q}(t))\right]_{t=a}^{t=b} - \int_a^b \phi(t)\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}(t,q,\dot{q})\right)dt$$

Since $\phi(b) = \phi(a) = 0$, by assumption, we obtain the following formula for the first variation:

$$\delta J(q)(\phi) = \int_a^b \left[\frac{\partial L}{\partial q} - \frac{d}{dt}\frac{\partial L}{\partial \dot{q}}\right]\phi(t)dt.$$

Therefore, in order for $\delta J(\phi)$ to vanish for all ϕ, the critical trajectory $q(t)$ must satisfy the *Euler-Lagrange equations*

$$\frac{\partial L}{\partial q} - \frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = 0. \tag{4.35}$$

4.B.2. Discrete Euler-Lagrange equations

Now consider two positions: q_0 and q_1, and a time step $h > 0$. We discretize a continuous Lagrangian $L(q,\dot{q})$ by assuming that q_1, q_0 are close together so that $\dot{q}$ can be approximated by $(q_1 - q_0)/h$. This allows us to define the following discrete Lagrangian

$$L_d(q_0, q_1; h) := \frac{h}{2}\left(L\left(q_0, \frac{q_1-q_0}{h}\right) + L\left(q_1, \frac{q_1-q_0}{h}\right)\right),$$

which approximates the action integral along a straight trajectory from q_0 to q_1. In the discrete Calculus of Variations, we replace a continuous curve $q(t)$ with a piece-wise linear curve determined by a sequence of points $\{q_k\}_{k=0}^N$ with h units of time required to go from q_k to q_{k+1}. We will now calculate the discrete action over this sequence by summing the discrete Lagrangian.

$$J_d = \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}; h).$$

We now vary the trajectory by $dq = \{dq_k\}_{k=0}^N$ with $dq_0 = dq_N = 0$ in order to fix the boundary points q_0, q_N. Note that we use dq rather than $\epsilon\phi$ to describe the variation because the discretized system has finite degrees of freedom. The variation of the discrete action can now be given as

$$\begin{aligned} dJ_d &= \sum_{j=1}^{N-1} \frac{\partial}{\partial \boldsymbol{x}_j} \left(\sum_{k=0}^{N-1} L_d(q_k, q_{k+1}; h) \right) dq_j \\ &= \sum_{k=0}^{N-1} \left[D_1 L_d(q_k, q_{k+1}; h) dq_k + D_2 L_d(q_k, q_{k+1}; h) dq_{k+1} \right] \end{aligned}$$

Recall that each $\boldsymbol{x}_j = (q_{j1}, \ldots, q_{jn})$ is a point in n-dimensional space, so that $\partial/\partial \boldsymbol{x}_j, D_1, D_2$ are actually n-vectors of partial derivative operators. Rearranging the above sum (this corresponds to the integration by parts step in the continuous case) we obtain

$$dJ_d = \sum_{k=1}^{N-1} \left[D_2 L_d(q_{k-1}, q_k; h) + D_1 L_d(q_k, q_{k+1}; h) \right] dq_k.$$

If we require that the variation of the action is 0 for all dq_k, then we obtain the discrete Euler-Lagrange equations

$$D_2 L_d(q_{k-1}, q_k; h) + D_1 L_d(q_k, q_{k+1}; h) = 0, \quad k = 1, \ldots, N-1. \tag{4.36}$$

4.C. Runge-Kutta method

Using the Fourth order Runge-Kutta method (RK4) [179], we can solve equation (4.9) for Euclidean space or (4.11) for spherical. There are many variations of the fourth order method, and they all use four approximations to the slope. We will use the following slope approximations to estimate the slope at some time t, using a time step Δt.

$$\begin{aligned} k_1 &= f(q_t, t) = \frac{dq_t}{dt} \\ k_2 &= f\left(q_t + k_1, t + \frac{\Delta t}{2}\right) \\ k_3 &= f\left(q_t + k_2 \frac{\Delta t}{2}, t + \frac{\Delta t}{2}\right) \\ k_4 &= f(q_t + k_3 \Delta t, t + \Delta t) \end{aligned} \tag{4.37}$$

Each of these slope estimates can be described verbally.

- k_1 is the slope at the beginning of the time step.
- If we use the slope k_1 to step halfway through the time step, then k_2 is an estimate of the slope at the midpoint. This slope proved to be more accurate than k_1 for making new approximations for q_t.

- If we use the slope k_2 to step halfway through the time step, then k_3 is another estimate of the slope at the midpoint.
- Finally, we use the slope, k_3, to step all the way across the time step (to $t + \Delta t$), and k_4 is an estimate of the slope at the endpoint.

We then use a weighted sum of these slopes to get our final estimate of $q(t + \Delta t)$:

$$q_{t+\Delta t} = q_t + \left(\frac{1}{6} k_1 + \frac{1}{3} k_2 + \frac{1}{3} k_3 + \frac{1}{6} k_4 \right) \Delta t \tag{4.38}$$

The lower Δt, the most accurate this approximation becomes, but it will be more computationally costly. RK4 method can be iterated until we get a list of coordinates q with length T, such that q_t is the coordinate of the vessel at time t. We are able to see how the drift affects vessel's true trajectory by plotting the path after T time steps, as shown in figure 4.16

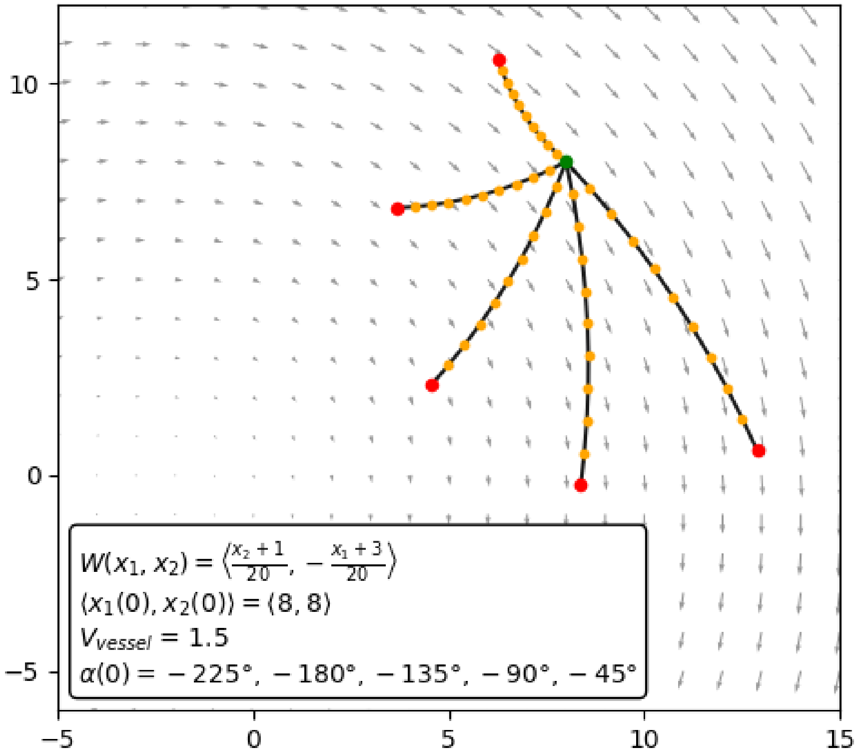


Figure 4.16: Example of RK4 application. Here we are using a very simple current, namely the circular vector field explained in section 4.3.2 with $(x_1, x_2) = (8, 8)$ (green dot), and various initial shooting angles α. Red dots are the end points computed with RK4 at $t = 4$, and orange dots are the way points generated each $\Delta t = 0.4$.

4.D. Code

The present research on weather routing was conducted using the Python programming language. The implementation of the variational method utilized in this study is publicly available for reference at `https://github.com/daniprec/hybrid_ivp`. It is important to emphasize that this research is an ongoing effort, and as such, results may differ from those presented in this thesis. Figure 4.17 offers a simplified depiction of the class diagram employed throughout this work, illustrating the general behavior of the code.

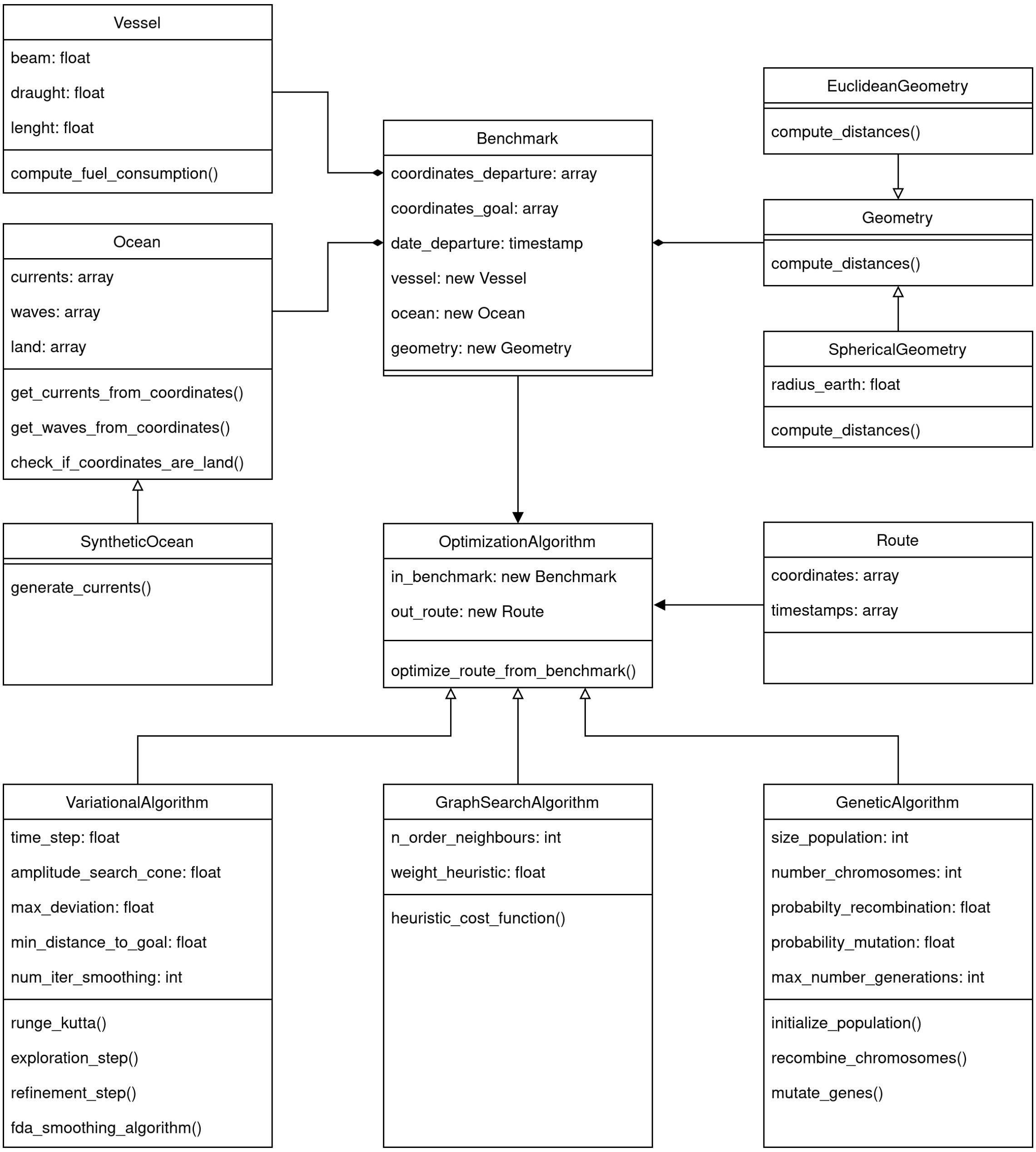


Figure 4.17: Basic representation of the class diagram for the weather routing code, implemented in Python.

5. CONCLUSIONS

5.1. Further work

We end up this thesis by summarising results and suggesting a few challenges. Several avenues are open for further work. Here we discuss only several promising ones, following our three core chapters.

5.1.1. Tuna biomass estimation

The purpose of Chapter 2 was to present a new pipeline for estimating tuna biomass aggregated at DFADs, named TUN-AI . The pipeline trains numerous ML models that handle distinct tasks related to fisheries operations using echo-sounder buoy, FAD logbook, and oceanographic data. We tested the performance for both classification and regression tasks, as well as the relative influence of integrating different data sources on model performance, to discover the most accurate model. The best classification model uses a 3-day window of echo-sounder data, oceanography, and position/time derived features. With a F_1-score of 0.925, this model can estimate if tuna biomass was greater than or less than 10t. When predicting tuna biomass directly, the best model (GB) has an error (MAE) of 21.6t and a relative error (SMAPE) of 29.5% when examined over sets. When supplemented with oceanographic and position-derived information, all models results improved, emphasising the usefulness of these variables when employing echo-sounder buoy data.

In the case of classification models, the confusion matrices in Figure 2.6 revealed that the majority of situations where the model misclassified the tuna aggregation size occurred when biomass estimations were medium ($10t \leq y < 30t$) or high ($y \geq 30t$). When we examined the regression models, we discovered that as the latter grew, estimated tuna biomass tended to be lower than observed tuna biomass (Figure 2.8). This might be attributed to a variety of circumstances, including: first, catches of more than 100t were relatively rare (315 events in our data, 8.1%), and thus the model did not have enough examples to properly learn from them; second, buoys can only estimate the biomass of tuna within the echo-sounder beam, and in tuna aggregations of more than 100t, the entire school is unlikely to be under the buoy at the same time. Furthermore, in huge schools of tuna, the echo-sounder signal may saturate, causing the biomass estimations produced by the echo-sounder buoy to be underestimated. To address this issue, it may be worthwhile to employ specialised models that can be altered based on whether aggregations are expected to be small or big. It is also worth mentioning that fisherman do not cast their nets at random buoys, but rather depending on the biomass assessment supplied to them, and so may be biassed towards buoys with larger biomass predictions. This might be another reason why our ML models overestimated the observed tuna biomass when its values exceeded 30t in the ternary classification and 100t in the regression tasks. Future research on the factors that influence fishermen's decisions to visit a buoy may shed more light on this topic. This propensity to underestimate should be considered when utilising information produced from echo-sounder buoys for stock assessments [58], albeit continuous underestimating should have little influence on patterns existing in the temporal series.

Tuna's pelagic and migratory tendencies make it a difficult species to research using traditional methods, since only a tiny portion of its habitat can be studied in real time at any given time. DFADs equipped with high-tech echo-sounder buoys, such as those employed in the current study, can, on the other hand, be utilised as floating open-ocean sampling stations, acquiring constant and up-to-date information from numerous sensors. As seen above, and as noted by other authors [62], while the information produced by the echo-sounder is important on its own, it still requires substantial cleaning and filtering prior to use. These preliminary processes can minimise inaccuracies caused by buoy measurements conducted on board or on land, but the ML algorithms utilised here go a step further in processing the acoustic data to accurately estimate tuna biomass beneath any particular echo-sounder buoy. In this regard, acoustic signals can offer information about the presence and behaviour of tuna near DFADs for a fraction of the expense associated with research expeditions of the same magnitude. As previously stated, this sort of data might be utilised for fishery-independent abundance indices [59], [180], [181], enhancing species distribution knowledge, or better understanding the causes driving tuna aggregation and disaggregation processes at DFADs [58], [60], [136]. The current work is a significant step in this approach since it is the first to correctly analyse the performance of several ML models utilising accurate ML methods and vast quantities of data to train and test each model. This has enabled TUN-AI to perform the most complicated tasks, such as directly predicting the amount of tuna aggregated to the DFAD with high degrees of accuracy. As demonstrated here, when the massive data provided by echo-sounder buoys attached to DFADs is enriched with remote-sensing data on ocean conditions and trained with reliable ground-truth data, ML proves a powerful tool for uncovering previously hidden patterns in these datasets, potentially expanding our understanding of pelagic species.

Lastly, it is worth mentioning that the biomass predictions can also be improved by designing DL architectures that are able to extract more information from the data. Figure 5.1 shows one architecture that we were designing at the moment of writing this thesis. We use convolutional neural networks to extract the correlation between different layer depths, and feed their outputs to recurrent neural networks (LSTM) that can extrapolate temporal patterns. Thus, the model learns from both spatial and temporal features in the data.

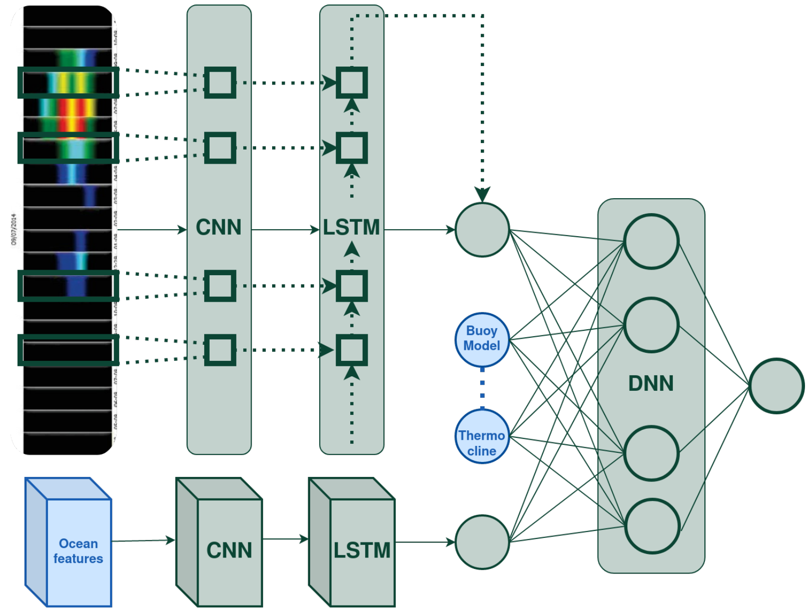


Figure 5.1: Proposed Deep Learning (DL) architecture to improve tuna biomass predictions. It combines convolutional and recurrent neural networks (CNN and LSTM respectively) to extract the most amount of information from spatial correlations between layer depths and temporal patterns.

5.1.2. Tuna dynamics

In Chapter 3 we applied the ML pipeline developed in Chapter 2, Tun-AI , to examine the temporal trends of tuna schools' association to drifting objects. Metrics commonly used in the literature were altered to account for the fact that the complete tuna aggregation under the DFAD was examined, using a binary output (tuna absent or present). The median time it took tuna to colonise the DFADs for the first time varied between 25 and 43 days, depending on the ocean, with the Pacific Ocean having the longest soak and colonisation time frames. Continuous Residence Time (CRT) for tuna schools was often shorter than Continuous Absence Time (CAT) (median values between 5-7 days and 9-11 days, respectively), consistent with earlier research. Two novel metrics, Aggregation's Continuous Residence Time (aCRT) and aCAT, were computed using regression output to get deeper insight into the symmetry of the aggregation process. The time it took for the tuna school to leave from the DFADs was not considerably longer than the time it took for the aggregation to develop throughout all oceans.

While the tuna dynamics study in Chapter 3 concentrated on the time patterns of tuna aggregation to DFADs, future research might focus on the spatial dynamics at work. Moreno, Dagorn, Sancho *et al.* [131] interviewed fishing masters who claimed that the departure of tuna schools from the DFADs was frequently connected to changes in currents or the FAD drift trajectory, and that variations in the surrounding environment, such as temperature, might also cause tuna to leave the DFAD. Examining the oceanographic background around DFADs during aggregation and disaggregation processes might help determine whether this is happening. Similarly, determining if local DFAD density affects CT, aCRT, or aCAT might help determine appropriate DFAD utilisation for purse-seine fleets. Though there are inherent difficulties in using data from echo-sounder buoys attached to DFADs, our study of Chapter 3 is an excellent example of how this information, combined with DS filtering and processing techniques, can give a low-cost tool for effectively providing insight on tuna biology and behaviour.

5.1.3. Weather routing

Based on the findings presented in Chapter 4, it is clear that weather routing algorithms can have a significant impact on the shipping industry. The use of these optimization methods can lead to significant improvements in vessel performance, such as reduced voyage times and lower operating costs. One strength of weather routing lies in the availability of a diverse range of optimization techniques. In our research, we have identified three distinct approaches and discovered unique advantages associated with each. The employment of the $A^\star$graph search method results in a reduction in fuel consumption, whereas the use of the Hybrid Search (HS) algorithm is superior in identifying routes that take less time, while genetic algorithm shows a good balance between fuel consumption and travel time. This broad spectrum of options is of significant benefit to shipping companies, as it enables them to prioritize specific advantages according to their business objectives.

However, it is important to note that the benefits of weather routing can be further amplified when combined with other strategies, such as slow steaming practices. Slow steaming is a technique that involves reducing vessel speed in order to decrease fuel consumption and emissions [86], [87]. When used in combination with weather routing algorithms, slow steaming can lead to even greater fuel savings and environmental benefits. Future research could focus on optimizing the balance between slow steaming and weather routing in order to maximize the benefits of both strategies.

Another emerging technology that has the potential to improve efficiency in shipping operations is the use of sails [88]. While traditional sailing vessels have largely been replaced by motorized ships, recent advancements in sail technology have led to renewed interest in the use of sails for commercial shipping. By pairing sails with slow steaming, shipping companies can reduce their reliance on fossil fuels and improve the sustainability of their operations, while also potentially reducing costs. Weather routing as a significant application on this scenarios, as algorithms can account for the influence of wind on the sails.

Furthermore, it is possible to enhance our optimization algorithms by integrating the safety of the voyage as a target parameter. This can be achieved by considering the avoidance of frontal waves, which have been demonstrated to result in vessel capsizing. Such an extension would provide an additional layer of protection to vessels and their occupants, ultimately leading to safer and more efficient maritime transportation.

Throughout this study on weather routing, a simple consumption model was utilized, which may benefit from further refinement. Although complex models tend to be computationally intensive, the development of lightweight ML models that leverage consumption data could provide reliable estimations of fuel consumed with reduced computational cost [182]–[184]. Such models have the potential to enhance the accuracy of fuel consumption predictions, allowing for more informed decisions to be made with regards to vessel operations and overall efficiency.

The incorporation of weather routing algorithms into shipping operations has shown the potential to achieve notable advancements in fuel efficiency and environmental sustainability. Nevertheless, the shipping industry is confronted with substantial safety concerns that must be addressed. Notably, weather routing algorithms have the ability to account for safety considerations as a part of their cost function. The assessment of safety concerns can be accomplished through various measures, such as monitoring the ship's average roll speed, which can result from the impact of waves hitting its sides, or the intensity of the wind, which can serve as an indicator of hazardous weather conditions.

In conclusion, the findings presented in this thesis demonstrate the significant potential of weather routing algorithms to improve the efficiency and sustainability of the shipping industry. By combining these algorithms with other strategies such as slow steaming and sails, shipping companies can reduce their costs and environmental impact. Continued research and development in this area is crucial in order to realize these benefits and ensure a more sustainable future for the shipping industry.

5.2. Crossover applications

This industrial thesis has researched two topics that are integral components of the blue economy, namely sustainable fishing and weather routing. Both areas have been approached through the application of Data Science (DS) methods and, in some instances, Machine Learning (ML) implementations. Furthermore, these two fields of study share common data sources, with weather routing reliant on oceanographic data, which is also significant in fish estimation algorithms. The overlap between sustainable fishing and weather routing extends beyond these areas of commonality. As a conclusion to this thesis, we have compiled a list of crossover applications that have been proposed and, on some occasions, implemented to enhance the performance of industrial procedures.

5.2.1. Applications of sustainable fishing to weather routing

The relevance of data obtained from Drifting Fish Aggregating Devices (DFADs) extends beyond just fish estimation algorithms and is also highly applicable to weather routing. The deployment of buoys equipped with an accelerometer enables the precise measurement of wave motion, while the daily GPS reports detailing the position of the buoys can provide valuable information on ocean currents in specific regions. This data can be utilized to generate more accurate weather forecasts [185], which in turn improves the impact of weather routing algorithms.

5.2.2. Applications of weather routing to sustainable fishing

Route optimization is a concept that already exists in the fishing industry [186], [187], and it involves using the predicted locations of fish populations to inform the route that a fishing vessel follows. This journey planning can be improved with weather routing. By knowing where the largest concentrations of fish are likely to be, a vessel can use weather routing to efficiently navigate to those areas and maximize their catch, while reducing operation costs.

Additionally, weather routing can be used to avoid areas with adverse meteorological conditions that may be harmful to the vessel or interfere with fishing operations. Optimization algorithms could also avoid areas where fish populations are low or where overfishing is a concern. By using weather routing to steer clear of these areas, fishing vessels can reduce their impact on fish populations and help to ensure that they are not contributing to overfishing.

Overall, the combination of weather routing and tuna biomass predictions can help fishing vessels to maximize their catch while also operating safely and efficiently.

BIBLIOGRAPHY

[1] D. Precioso and D. Gómez-Ullate, 'Sustainable Fishing: applying Data Science to the Ecological Trap Hypothesis,' *Actas de las Jornadas de Investigación Predoctoral en Ingeniería Informática*, p. 21, 2021.

[2] D. Precioso *et al.*, 'TUN-AI: Tuna biomass estimation with Machine Learning models trained on oceanography and echosounder FAD data,' *Fisheries Research*, vol. 250, no. February, p. 106 263, 2022. DOI: `10.1016/j.fishres.2022.106263`.

[3] D. Gómez-Ullate *et al.*, 'Smart Shipping: Optimización de rutas marítimas en tiempo real,' *61 Congreso de Ingeniería Naval e Industria Marítima*, 2022.

[4] M. Navarro-García *et al.*, 'How do tuna schools associate to dFADs? A study using echo-sounder buoys to identify global patterns,' *arXiv preprint arXiv:2207.07049*, 2022.

[5] D. Precioso, R. Milson, L. Bu, V. Menchions and D. Gómez-Ullate, 'Hybrid Search method for Zermelo's navigation problem,' 2023.

[6] D. Precioso and D. Gómez-Ullate, 'Thresholding Methods in Non-Intrusive Load Monitoring,' *The Journal of Supercomputing*, 2023.

[7] D. Precioso and D. Gómez-Ullate, 'Non-Intrusive Load Monitoring using Multi-Output CNNs,' in *2021 IEEE Madrid PowerTech*, 2021, pp. 1–6. DOI: `10.1109/PowerTech46648.2021.9494943`.

[8] I. Barbeito *et al.*, 'Effectiveness of non-pharmaceutical interventions in nine fields of activity to decrease SARS-CoV-2 transmission (Spain, September 2020-May 2021),' *Frontiers in Public Health*, 2023.

[9] A. Ruiz-Zafra *et al.*, 'NeoCam: An edge-cloud platform for non-invasive real-time monitoring in neonatal intensive care units,' *IEEE Journal of Biomedical and Health Informatics*, pp. 1–12, 2023. DOI: `10.1109/JBHI.2023.3240245`.

[10] United Nations, 'The Oceans Economy: Opportunities and Challenges for Small Island Developing States,' *United Nations Conference on Trade and Development*, 2012.

[11] S. Smith-Godfrey, 'Defining the blue economy,' *Maritime affairs: Journal of the national maritime foundation of India*, vol. 12, no. 1, pp. 58–64, 2016.

[12] United Nations, 'Technical report by the Bureau of the United Nations Statistical Commission on the process of the development of an indicator framework for the goals and targets of the post-2015 development agenda,' *United Nations Conference on Trade and Development*, 2015.

[13] United Nations, *Goal 14 Conserve and sustainably use the oceans, seas and marine resources for sustainable development*, 2015.

[14] J. S. Golden, J. Virdin, D. Nowacek, P. Halpin, L. Bennear and P. G. Patil, 'Making sure the blue economy is green,' *Nature Ecology & Evolution*, vol. 1, no. 2, pp. 1–3, 2017.

[15] R. C. Brears, *Developing the Blue Economy*. Springer, 2021.

[16] M. Sarkar, P. Chowdhury, J. Jarin and A. Islam, 'Machine learning to understand marine ecosystems and harness the blue economy,' *Academia Letters*, p. 2, 2021.

[17] R. Vinuesa *et al.*, 'The role of artificial intelligence in achieving the Sustainable Development Goals,' *Nature communications*, vol. 11, no. 1, pp. 1–10, 2020.

[18] N. Lambert, J. Turner and A. Hamflett, *Technology and the blue economy: from autonomous shipping to big data*. Kogan Page Publishers, 2019.

[19] V. Dhar, 'Data science and prediction,' *Communications of the ACM*, vol. 56, no. 12, pp. 64–73, 2013.

[20] K. D. Foote, 'A Brief Story of Data Science,' *Dataversity*, 2021.

[21] J. W. Tukey, 'The Future of Data Analysis,' *Institute of Mathematical Statistics*, 1962.

[22] P. Naur, *Concise survey of computer methods*. Petrocelli Books, 1974.

[23] T. Davenport, 'Data Scientist: The Sexiest Job of the 21st Century,' *Harvard Business Review*, 2012.

[24] A. L. Samuel, 'Some studies in machine learning using the game of checkers,' *IBM Journal of research and development*, vol. 11, no. 6, pp. 601–617, 1967.

[25] G. Slava, 'How the Computer Got Its Revenge on the Soviet Union,' *Nautilus*, 2015.

[26] M. P. Deisenroth, A. A. Faisal and C. S. Ong, *Mathematics for machine learning*. Cambridge University Press, 2020.

[27] T. M. Mitchell, *Machine Learning*. 1997, p. 414.

[28] M. Mohri, A. Rostamizadeh and A. Talwalkar, *Foundations of Machine Learning*. 2012.

[29] P. Cunningham, M. Cord and S. J. Delany, 'Supervised learning,' in *Machine learning techniques for multimedia*, Springer, 2008, pp. 21–49.

[30] S. Russell and P. Norvig, 'Artificial Intelligence: A modern approach,' *Prentice Hall*, 2010.

[31] Y. Lecun, Y. Bengio and G. Hinton, *Deep learning*. Nature Publishing Group, May 2015, vol. 521, pp. 436–444. DOI: `10.1038/nature14539`.

[32] H.-K. Chiou, G.-H. Tzeng and D.-C. Cheng, ‘Evaluating sustainable fishing development strategies using fuzzy MCDM approach,’ *Omega*, vol. 33, no. 3, pp. 223–234, 2005. DOI: `https://doi.org/10.1016/j.omega.2004.04.011`.

[33] G. Heal and W. Schlenker, ‘Sustainable fisheries,’ *Nature*, vol. 455, no. 7216, pp. 1044–1045, 2008.

[34] B. Leroy *et al.*, ‘Lessons learned from implementing three, large-scale tuna tagging programmes in the western and central Pacific Ocean,’ *Fisheries Research*, vol. 163, pp. 23–33, 2015.

[35] J. Hampton, ‘Estimates of tag-reporting and tag-shedding rates in a large-scale tuna tagging experiment in the western tropical Pacific Ocean,’ *Oceanographic Literature Review*, vol. 11, no. 44, p. 1346, 1997.

[36] J. Hampton, ‘Natural mortality rates in tropical tunas: Size really does matter,’ *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 57, no. 5, pp. 1002–1010, 2000.

[37] C. Brownie, *Statistical inference from band recovery data: a handbook*. US Department of the Interior, Fish and Wildlife Service, 1978, vol. 131.

[38] J. Hampton and D. A. Fournier, ‘A spatially disaggregated, length-based, age-structured population model of yellowfin tuna (Thunnus albacares) in the western and central Pacific Ocean,’ *Marine and Freshwater Research*, vol. 52, no. 7, pp. 937–963, 2001.

[39] G. Moreno, J. Murua and V. Restrepo, ‘The use of echo-sounder buoys in purse seine fleets fishing with DFADs in the eastern Pacific Ocean,’ in *IATTC-7th Meeting of the Scientific Advisory Committee*, 2016.

[40] J. Lopez, G. Moreno, I. Sancristobal and J. Murua, ‘Evolution and current state of the technology of echo-sounder buoys used by Spanish tropical tuna purse seiners in the Atlantic, Indian and Pacific Oceans,’ en, *Fisheries Research*, vol. 155, pp. 127–137, May 2014. DOI: `10.1016/j.fishres.2014.02.033`.

[41] Office of Protected Resources, *Fishing Gear: Fish Aggregating Devices*, `https://www.fisheries.noaa.gov/national/bycatch/fishing-gear-fish-aggregating-devices`, 2017.

[42] M. J. Kingsford, ‘Biotic and abiotic structure in the pelagic environment: Importance to small fishes,’ *Bulletin of Marine Science*, vol. 53, no. 2, pp. 393–415, 1993.

[43] A. Fonteneau, P. Pallarés and R. Pianet, ‘A worldwide review of purse seine fisheries on FADs,’ en, *Regional syntheses*, p. 21, 2000.

[44] J. J. Castro and A. T. Santana-Ortega, ‘A general theory on fish aggregation to floating objects: An alternative to the meeting point hypothesis,’ *Reviews in Fish Biology and Fisheries*, vol. 11, no. 3, p. 24, 2002. DOI: `10.1023/A:1020302414472`.

[45] A. Maufroy, E. Chassot, R. Joo and D. M. Kaplan, 'Large-Scale Examination of Spatio-Temporal Patterns of Drifting Fish Aggregating Devices (dFADs) from Tropical Tuna Fisheries of the Indian and Atlantic Oceans,' en, *PLOS ONE*, vol. 10, no. 5, G. Hays, Ed., e0128023, Apr. 2015. DOI: `10.1371/journal.pone.0128023`.

[46] G. Wain, L. Guéry, D. M. Kaplan, D. Gaertner and R. O'Driscoll, 'Quantifying the increase in fishing efficiency due to the use of drifting FADs equipped with echosounders in tropical tuna purse seine fisheries,' en, *ICES Journal of Marine Science*, vol. 78, no. 1, R. O'Driscoll, Ed., pp. 235–245, Apr. 2021. DOI: `10.1093/icesjms/fsaa216`.

[47] ISSF, 'Status of the World Fisheries for Tuna. Mar 2021,' *ISSF Technical Report 2021-10*, vol. March 2021, no. March, pp. 1–120, 2021.

[48] T. Fauvel *et al.*, 'Comparative study of the distribution of natural versus artificial drifting Fish Aggregating Devices (FADs) in the Western Indian Ocean,' *Working document for IOTC. Indian Ocean Tuna Commission*, 2009.

[49] Staff, *A brief history of FADs*, `https://www.lahainanews.com/sports/local-sports/2020/06/18/a-brief-history-of-fads/`, 2020.

[50] Oppian, *Halieutica or Fishing*, trans. by A. W. Mair. Loeb Classical Library, 1928.

[51] P. R. Greenblatt, 'Associations of tuna with flotsam in the eastern tropical Pacific,' *Fishery Bulletin*, vol. 77, no. 1-2, p. 147, 1979.

[52] J.-P. Hallier and J. I. Parajua, 'Review of tuna fisheries on floating objects in the Indian Ocean,' in *Proc Int Workshop on Fishing for Tunas Associated with Floating Objects, La Jolla, CA*, 1992, pp. 195–221.

[53] A. Maufroy *et al.*, 'Massive increase in the use of drifting Fish Aggregating Devices (dFADs) by tropical tuna purse seine fisheries in the Atlantic and Indian oceans,' *ICES Journal of Marine Science*, vol. 74, no. 1, pp. 215–225, Oct. 2016. DOI: `10.1093/icesjms/fsw175`. eprint: `https://academic.oup.com/icesjms/article-pdf/74/1/215/31245298/fsw175.pdf`.

[54] T. K. Davies, C. C. Mees and E. J. Milner-Gulland, 'The past, present and future use of drifting fish aggregating devices (FADs) in the Indian Ocean,' *Marine Policy*, vol. 45, pp. 163–170, Mar. 2014. DOI: `10.1016/j.marpol.2013.12.014`.

[55] J. Lopez and G. P. Scott, 'The use of FADs in tuna fisheries,' *European Union*, vol. 1, no. February, p. 70, 2014.

[56] D. D. Molina, P. Pallares, J. J. Areso and J. Ariz, 'Statistics of the purse seine spanish fleet in the Indian Ocean (1984-2002),' *IOTC Proceedings no. 2*, vol. 6, no. 6, pp. 115–128, 2003.

[57] Y. Baidai, L. Dagorn, M. J. Amande, D. Gaertner and M. Capello, 'Machine learning for characterizing tropical tuna aggregations under Drifting Fish Aggregating Devices (DFADs) from commercial echosounder buoys data,' en, *Fisheries Research*, vol. 229, p. 105 613, Apr. 2020. doi: 10.1016/j.fishres.2020.105613.

[58] J. Santiago, J. Lopez, G. Moreno, H. Murua, I. Quincoces and M. Soto, 'Towards a Tropical Tuna Buoy-Derived Abundance Index (TT-BAI),' *Collective Volume of Scientific Papers ICCAT*, vol. 72, pp. 714–724, 2016.

[59] J. Santiago *et al.*, 'A Novel Index of Abundance of Juvenile Yellowfin Tuna in the Atlantic Ocean Derived from Echosounder Buoys,' *Collective Volume of Scientific Papers ICCAT*, vol. 76, pp. 321–343, 2020.

[60] J. Lopez, G. Moreno, G. Boyra and L. Dagorn, 'A model based on data from echosounder buoys to estimate biomass of fish species associated with fish aggregating devices,' *Fishery Bulletin*, vol. 114, no. 2, pp. 166–178, Feb. 2016. doi: 10.7755/FB.114.2.4.

[61] L. Escalle *et al.*, 'Report on preliminary analyses of FAD acoustic data,' en, *Western and Central Pacific Fisheries Commission*, vol. 53, no. 9, p. 17, 2019.

[62] B. Orue, J. Lopez, G. Moreno, J. Santiago, M. Soto and H. Murua, 'Aggregation process of drifting fish aggregating devices (DFADs) in the Western Indian Ocean: Who arrives first, tuna or non-tuna species?' en, *PLOS ONE*, vol. 14, no. 1, D. Hyrenbach, Ed., e0210435, Apr. 2019. doi: 10.1371/journal.pone.0210435.

[63] J. Lopez, 'Environmental preferences of tuna and non-tuna species associated with drifting fish aggregating devices (DFADs) in the Atlantic Ocean, ascertained through fishers' echo-sounder buoys,' en, *Deep-Sea Research II*, p. 12, 2017.

[64] J. N. Druon, E. Chassot, H. Murua and J. Lopez, 'Skipjack tuna availability for purse seine fisheries is driven by suitable feeding habitat dynamics in the Atlantic and Indian Oceans,' *Frontiers in Marine Science*, vol. 4, no. OCT, p. 315, Oct. 2017. doi: 10.3389/fmars.2017.00315.

[65] K. M. Schaefer, D. W. Fuller and B. A. Block, 'Movements, behavior, and habitat utilization of yellowfin tuna (Thunnus albacares) in the northeastern Pacific Ocean, ascertained through archival tag data,' *Marine Biology*, vol. 152, no. 3, pp. 503–525, Sep. 2007. doi: 10.1007/s00227-007-0689-x.

[66] P. Fréon and L. Dagorn, 'Review of fish associative behaviour: toward a generalisation of the meeting point hypothesis,' *Reviews in Fish Biology and Fisheries*, vol. 10, no. 2, pp. 183–207, 2000. doi: 10.1023/a:1016666108540.

[67] T. Dempster and M. Taquet, 'Fish aggregation device (FAD) research: gaps in current knowledge and future directions for ecological studies,' *Reviews in Fish Biology and Fisheries*, vol. 14, pp. 21–42, 2004.

[68] M. Hall, C. Lennert-Cody, M. Garcia and P. Arenas, 'Characteristics of floating objects and their attractiveness for tunas,' in *Proceedings of the International Workshop on the Ecology and Fisheries for Tunas Associated with Floating Objects*, 1992, pp. 396–446.

[69] F. Marsac, A. Fonteneau and F. Ménard, 'Drifting FADs used in tuna fisheries: an ecological trap?' en, *Biology and behaviour of pelagic fish aggregations*, no. July 2015, p. 17, 2000.

[70] J. Hallier and D. Gaertner, 'Drifting fish aggregation devices could act as an ecological trap for tropical tuna species,' en, *Marine Ecology Progress Series*, vol. 353, pp. 255–264, Apr. 2008. DOI: `10.3354/meps07180`.

[71] L. Dagorn, K. N. Holland, V. Restrepo and G. Moreno, 'Is it good or bad to fish with FADs? What are the real impacts of the use of drifting FADs on pelagic marine ecosystems?' en, *Fish and Fisheries*, vol. 14, no. 3, pp. 391–415, Apr. 2012. DOI: `10.1111/j.1467-2979.2012.00478.x`.

[72] Y. Baidai, L. Dagorn, M. J. Amande, D. Gaertner and M. Capello, 'Mapping Tuna Occurrence Under Drifting Fish Aggregating Devices From Fisher's Echosounder buoys in Indian Ocean,' in *21st Working Party on Tropical Tunas (WPTT)*, 2019.

[73] L. Dagorn, K. N. Holland and J. Filmalter, 'Are drifting FADs essential for testing the ecological trap hypothesis?' *Fisheries Research*, vol. 106, no. 1, pp. 60–63, Oct. 2010. DOI: `10.1016/j.fishres.2010.07.002`.

[74] W. Shao, P. Zhou and S. K. Thong, 'Development of a novel forward dynamic programming method for weather routing,' *Journal of Marine Science and Technology*, vol. 17, pp. 239–251, May 2012. DOI: `10.1007/s00773-011-0152-z`.

[75] T. P. Zis, H. N. Psaraftis and L. Ding, 'Ship weather routing: A taxonomy and survey,' *Ocean Engineering*, vol. 213, p. 107 697, Oct. 2020. DOI: `10.1016/J.OCEANENG.2020.107697`.

[76] IMO, 'Initial IMO Strategy on Reduction of Ghg Emissions from Ships,' *Resolution MEPC.304/(72)*, 2018.

[77] EC, 'Reducing emissions from the shipping sector,' *European Comission*, 2021.

[78] J. Barreiro, S. Zaragoza and V. Diaz-Casas, 'Review of ship energy efficiency,' *Ocean Engineering*, vol. 257, p. 111 594, Aug. 2022. DOI: `10.1016/J.OCEANENG.2022.111594`.

[79] J. Huffmeier and M. Johanson, 'State-of-the-art methods to improve energy efficiency of ships,' *Journal of Marine Science and Engineering 2021, Vol. 9, Page 447*, vol. 9, p. 447, Apr. 2021. DOI: `10.3390/JMSE9040447`.

[80] K. Wang, X. Yan, Y. Yuan, X. Jiang, X. Lin and R. R. Negenborn, 'Dynamic optimization of ship energy efficiency considering time-varying environmental factors,' *Transportation Research Part D: Transport and Environment*, 2018.

[81] L. C. Law, B. Foscoli, E. Mastorakos and S. Evans, ‘A comparison of alternative fuels for shipping in terms of lifecycle energy and cost,’ *Energies*, vol. 14, no. 24, p. 8502, 2021.

[82] E. B. Besikci, T. Kececi, O. Arslan and O. Turan, ‘An application of fuzzy-AHP to ship operational energy efficiency measures,’ *Ocean Engineering*, vol. 121, pp. 392–402, 2016. DOI: `https://doi.org/10.1016/j.oceaneng.2016.05.031`.

[83] United Nations, ‘Review of Maritime Transport,’ *United Nations Conference on Trade and Development*, 2019.

[84] M. Bentin, D. Zastrau, M. Schlaak, D. Freye, R. Elsner and S. Kotzur, ‘A new routing optimization tool-influence of wind and waves on fuel consumption of ships with and without wind assisted ship propulsion systems,’ *Transportation Research Procedia*, vol. 14, pp. 153–162, May 2016. DOI: `10.1016/j.trpro.2016.05.051`.

[85] Z. Zhang and X. M. Li, ‘Global ship accidents and ocean swell-related sea states,’ *Natural Hazards andEarth System Sciences, 17(11)*, 2017.

[86] P. Cariou, ‘Is slow steaming a sustainable means of reducing CO2 emissions from container shipping?’ *Transportation Research Part D: Transport and Environment*, vol. 16, no. 3, pp. 260–264, 2011. DOI: `https://doi.org/10.1016/j.trd.2010.12.005`.

[87] H. N. Psaraftis and C. A. Kontovas, ‘Balancing the economic and environmental performance of maritime transportation,’ *Transportation Research Part D: Transport and Environment*, vol. 15, no. 8, pp. 458–462, 2010. DOI: `https://doi.org/10.1016/j.trd.2010.05.001`.

[88] C. D. Beukelaer, ‘Tack to the future: is wind propulsion an ecomodernist or degrowth way to decarbonise maritime cargo transport?’ *Climate Policy*, vol. 22, no. 3, pp. 310–319, 2022. DOI: `10.1080/14693062.2021.1989362`. eprint: `https://doi.org/10.1080/14693062.2021.1989362`.

[89] I. S. Seddiek and N. R. Ammar, ‘Harnessing wind energy on merchant ships: Case study flettner rotorsonboard bulk carriers,’ *Environmental Science and Pollution Research*, 2021.

[90] P. Zhou, H. Wang and Z. Guan, ‘Ship weather routing based on grid system and modified genetic algorithm,’ in *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, 2019, pp. 647–652. DOI: `10.1109/ISIE.2019.8781243`.

[91] J. Uranga *et al.*, ‘Detecting the presence-absence of bluefin tuna by automated analysis of medium-range sonars on fishing vessels,’ *PloS one*, vol. 12, no. 2, e0171382, 2017.

[92] L. Breiman, ‘Random Forests,’ *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. DOI: `10.1023/A:1010933404324`.

[93] C. J. C. Burges, ‘A tutorial on support vector machines for pattern recognition,’ *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[94] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[95] J. R. Quinlan, ‘Improved use of continuous attributes in C4. 5,’ *Journal of artificial intelligence research*, vol. 4, pp. 77–90, 1996.

[96] D. W. Aha, D. Kibler and M. K. Albert, ‘Instance-based learning algorithms,’ *Machine learning*, vol. 6, no. 1, pp. 37–66, 1991.

[97] L. Mannocci, Y. Baidai, F. Forget, M. T. Tolotti, L. Dagorn and M. Capello, ‘Machine learning to detect bycatch risk: Novel application to echosounder buoys data in tuna purse seine fisheries,’ *Biological Conservation*, vol. 255, no. February, p. 109 004, 2021. DOI: `10.1016/j.biocon.2021.109004`.

[98] T. Hastie and R. Tibshirani, ‘Generalized additive models: Some applications,’ *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 371–386, 1987.

[99] B. Orue *et al.*, ‘Using fishers’ echo-sounder buoys to estimate biomass of fish species associated with drifting fish aggregating devices in the Indian Ocean,’ *Revista de Investigación Marina*, vol. 26, no. 1, pp. 3–12, 2019.

[100] P. Lehodey, I. Senina and R. Murtugudde, ‘A spatial ecosystem and populations dynamics model (SEAPODYM) – Modeling of tuna and tuna-like populations,’ *Progress in Oceanography*, vol. 78, no. 4, pp. 304–318, 2008. DOI: `https://doi.org/10.1016/j.pocean.2008.06.004`.

[101] I. Senina, J. Sibert and P. Lehodey, ‘Parameter estimation for basin scale ecosystem linked population models of large pelagic predators: Application to skipjack tuna,’ *Progress in Oceanography*, vol. 78, no. 4, pp. 319–335, 2008.

[102] M. L. Ramos *et al.*, ‘Spanish FADs logbook: solving past issues, responding to new global requirements,’ *1st Ad-Hoc IOTC Working Group on FADs*, vol. 2017, no. April, pp. 1–24, 2017.

[103] G. Boyra, G. Moreno, B. Sobradillo, I. Pérez-Arjona, I. Sancristobal and D. A. Demer, ‘Target strength of skipjack tuna (Katsuwanus pelamis) associated with fish aggregating devices (FADs),’ *ICES Journal of Marine Science*, vol. 75, no. 5, pp. 1790–1802, Oct. 2018. DOI: `10.1093/ICESJMS/FSY041`.

[104] Global Monitoring and Forecasting Center, ‘Operational Mercator global ocean analysis and forecast system, E.U. Copernicus Marine Service Information,’ *Copernicus Marine Service (Accessed: 15th January 2021)*, 2018.

[105] J. M. Lellouche *et al.*, 'Recent updates to the Copernicus Marine Service global ocean monitoring and forecasting real-time 1g 12o high-resolution system,' *Ocean Science*, vol. 14, no. 5, pp. 1093–1126, 2018. DOI: `10.5194/os-14-1093-2018`.

[106] F. G. Forget *et al.*, 'Behaviour and vulnerability of target and non-target species at drifting fish aggregating devices (FADs) in the tropical tuna purse seine fishery determined by acoustic telemetry,' *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 72, no. 9, pp. 1398–1405, 2015. DOI: `10.1139/cjfas-2014-0458`.

[107] L. Dagorn, K. N. Holland and D. G. Itano, 'Behavior of yellowfin (Thunnus albacares) and bigeye (T. obesus) tuna in a network of fish aggregating devices (FADs),' *Marine Biology*, vol. 151, no. 2, pp. 595–606, Apr. 2007. DOI: `10.1007/s00227-006-0511-1`.

[108] D. R. Cox, 'The Regression Analysis of Binary Sequences,' *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, pp. 215–232, 1958. DOI: `10.1111/j.2517-6161.1958.tb00292.x`.

[109] H. Zou and T. Hastie, 'Regularization and variable selection via the elastic net,' *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005. DOI: `10.1111/j.1467-9868.2005.00503.x`.

[110] J. H. Friedman, 'Greedy function approximation: A gradient boosting machine,' *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001. DOI: `10.1214/aos/1013203451`.

[111] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System,' in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, New York, NY, USA: ACM, 2016, pp. 785–794. DOI: `10.1145/2939672.2939785`.

[112] F. Pedregosa *et al.*, 'Scikit-learn: Machine Learning in Python,' en, *Machine Learning in Python*, p. 6, 2011.

[113] H. A. Andrade, 'The relationship between the skipjack tuna (Katsuwonus pelamis) fishery and seasonal temperature variability in the south-western Atlantic,' *Fisheries Oceanography*, vol. 12, no. 1, pp. 10–18, Jan. 2003. DOI: `10.1046/J.1365-2419.2003.00220.X`.

[114] D. N. Maclennan, P. G. Fernandes and J. Dalen, 'A consistent approach to definitions and symbols in fisheries acoustics,' *ICES Journal of Marine Science*, vol. 59, no. 2, pp. 365–369, 2002. DOI: `10.1006/jmsc.2001.1158`.

[115] E. Josse, P. Bach and L. Dagorn, 'Simultaneous observations of tuna movements and their prey by sonic tracking and acoustic surveys,' *Hydrobiologia*, vol. 371, pp. 61–69, 1998.

[116] E. Josse and A. Bertrand, 'In situ acoustic target strength measurements of tuna associated with a fish aggregating device,' *ICES Journal of Marine Science*, vol. 57, no. 4, pp. 911–918, 2000.

[117] J. Simmonds and D. N. MacLennan, *Fisheries acoustics: theory and practice*. John Wiley & Sons, 2008.

[118] L. Dagorn *et al.*, 'Satellite-linked acoustic receivers to observe behavior of fish in remote areas,' *Aquatic Living Resources*, vol. 20, no. 4, pp. 307–312, 2007.

[119] M. T. Tolotti *et al.*, 'Association dynamics of tuna and purse seine bycatch species with drifting fish aggregating devices (FADs) in the tropical eastern Atlantic Ocean,' *Fisheries Research*, vol. 226, 2020. DOI: `10.1016/j.fishres.2020.105521`.

[120] I. Ohta and S. Kakuma, 'Periodic behavior and residence time of yellowfin and bigeye tuna associated with fish aggregating devices around Okinawa Islands, as identified with automated listening stations,' *Marine Biology*, vol. 146, no. 3, pp. 581–594, 2004. DOI: `10.1007/s00227-004-1456-x`.

[121] M. Robert, L. Dagorn, J. L. Deneubourg, D. Itano and K. Holland, 'Size-dependent behavior of tuna in an array of fish aggregating devices (FADs),' *Marine Biology*, vol. 159, no. 4, pp. 907–914, 2012. DOI: `10.1007/s00227-011-1868-3`.

[122] R. Govinden, L. Dagorn, J. Filmalter and M. Soria, 'Behaviour of Tuna associated with drifting fish aggregating devices (FADs) in the Mozambique Channel,' *IOTC proceedings. IOTC, Victoria*, vol. 22, 2010.

[123] T. Matsumoto, K. Satoh and M. Toyonaga, 'Behavior of skipjack tuna (Katsuwonus pelamis) associated with a drifting FAD monitored with ultrasonic transmitters in the equatorial central Pacific Ocean,' *Fisheries Research*, vol. 157, pp. 78–85, 2014. DOI: `10.1016/j.fishres.2014.03.023`.

[124] T. Matsumoto, K. Satoh, Y. Semba and M. Toyonaga, 'Comparison of the behaviour of skipjack (katsuwonus pelamis), yellowfin (thunnus albacares) and bigeye (t. obesus) tuna associated with drifting fads in the equatorial central pacific ocean,' *Fisheries Oceanography*, vol. 25, no. 6, pp. 565–581, Nov. 2016. DOI: `10.1111/fog.12173`.

[125] J. Scutt Phillips, T. Peatman, L. Escalle, B. Leroy and N. Smith, 'Electronic tagging for the mitigation of bigeye and yellowfin tuna juveniles by purse seine fisheries,' Aug. 2019.

[126] A. Diallo, Y. Baidai, L. Manocci and M. Capello, 'Towards the derivation of fisheries independent abundance indices for tropical tuna: Report on biomass estimates obtained from a multi-frequency echosounder buoy model (M3I+),' *IOTC-2019-WPTT21-54_Rev1*, 2019.

[127] Y. Baidai, L. Dagorn, M. J. Amandé, D. Gaertner, M. Capello and R. Proud, 'Tuna aggregation dynamics at Drifting Fish Aggregating Devices: a view through the eyes of commercial echosounder buoys,' *ICES Journal of Marine Science*, vol. 77, no. 7-8, pp. 2960–2970, 2020. DOI: `10.1093/icesjms/fsaa178`.

[128] M. Navarro-García, *cpsplines*, `https://github.com/ManuelNavarroGarcia/cpsplines`, version 0.1.2, 2021.

[129] P. Virtanen *et al.*, 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,' *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: `10.1038/s41592-019-0686-2`.

[130] L. Escalle, B. Muller, T. Vidal, S. Hare, P. Hamer and PNA Office, *Report on analyses of the 2016/2021 PNA FAD tracking programme (24July-10Aug) - Rev.01 | WCPFC Meetings*, Aug. 2021.

[131] G. Moreno, L. Dagorn, G. Sancho and D. Itano, 'Fish behaviour from fishers' knowledge: The case study of tropical tuna around drifting fish aggregating devices (DFADs),' *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 64, no. 11, pp. 1517–1528, 2007. DOI: `10.1139/F07-113`.

[132] R. Govinden, M. Capello, F. Forget, J. D. Filmalter and L. Dagorn, 'Behavior of skipjack (Katsuwonus pelamis), yellowfin (Thunnus albacares), and bigeye (T. obsesus) tunas associated with drifting fish aggregating devices (dFADs) in the Indian Ocean, assessed through acoustic telemetry,' *Fisheries Oceanography*, no. 00, pp. 1–14, 2021. DOI: `10.1111/fog.12536`.

[133] K. M. Schaefer and D. W. Fuller, 'Behavior of bigeye (Thunnus obesus) and skipjack (Katsuwonus pelamis) tunas within aggregations associated with floating objects in the equatorial eastern Pacific,' *Marine Biology*, vol. 146, no. 4, pp. 781–792, 2005. DOI: `10.1007/s00227-004-1480-x`.

[134] W. C. Chiang *et al.*, 'Fine-scale vertical movements and behavior of immature skipjack tuna (katsuwonus pelamis) off eastern taiwan,' *Journal of Marine Science and Technology (Taiwan)*, vol. 29, no. 2, pp. 207–219, 2021. DOI: `10.51400/2709-6998.1082`.

[135] P. Rodriguez-Tress *et al.*, 'Associative behavior of yellowfin Thunnus albacares, skipjack Katsuwonus pelamis, and bigeye tuna T. obesus at anchored fish aggregating devices (FADs) off the coast of Mauritius,' *Marine Ecology Progress Series*, vol. 570, pp. 213–222, Apr. 2017. DOI: `10.3354/meps12101`.

[136] G. Moreno, G. Boyra, I. Sancristobal, D. Itano and V. Restrepo, 'Towards acoustic discrimination of tropical tuna associated with Fish Aggregating Devices,' *PLOS ONE*, vol. 14, no. 6, e0216353, Jun. 2019. DOI: `10.1371/JOURNAL.PONE.0216353`.

[137] R. W. James, *Application of wave forecasts to marine navigation*. New York University, 1957.

[138] G. L. Hanssen and R. W. James, 'Optimum ship routing,' *The Journal of Navigation*, vol. 13, no. 3, pp. 253–272, 1960.

[139] H. Hagiwara, 'Weather routing of (sail-assisted) motor vessels,' *Delft University of Technology, Delft*, 1989.

[140] M.-I. Roh, 'Determination of an economical shipping route considering the effects of sea state for lower fuel consumption,' *International Journal of Naval Architecture and Ocean Engineering*, vol. 5, pp. 246–262, Jun. 2013. DOI: 10.2478/IJNAOE-2013-0130.

[141] Y.-h. H. Lin, M.-C. C. Fang and R. W. Yeung, 'The optimization of ship weather-routing algorithm based on the composite influence of multi-dynamic elements,' *Applied Ocean Research*, vol. 43, pp. 184–194, May 2013. DOI: 10.1016/j.apor.2013.07.010.

[142] G. J. Haltiner, H. D. Hamilton and G. Arnason, 'Minimal-time ship routing,' *Journal of Applied Meteorology (1962-1982)*, pp. 1–7, 1962.

[143] N. A. Papadakis and A. N. Perakis, 'Deterministic minimal time vessel routing,' *Operations Research*, vol. 38, pp. 426–438, May 1990. DOI: 10.1287/opre.38.3.426.

[144] S. J. Ferraro, D. M. de Diego and R. T. S. M. de Almagro, 'Parallel iterative methods for variational integration applied to navigation problems,' *IFAC-PapersOnLine*, vol. 54, pp. 321–326, Jan. 2021. DOI: 10.1016/J.IFACOL.2021.11.097.

[145] L. Walther, A. Rizvanolli, M. Wendebourg and C. Jahn, 'Modeling and optimization algorithms in ship weather routing,' *International Journal of e-Navigation and Maritime Economy*, vol. 4, pp. 31–45, May 2016. DOI: 10.1016/j.enavi.2016.06.004.

[146] R. Bellman, 'On the theory of dynamic programming,' *Proceedings of the national Academy of Sciences*, vol. 38, no. 8, pp. 716–719, 1952.

[147] D. Sidoti *et al.*, 'A Multiobjective Path-Planning Algorithm with Time Windows for Asset Routing in a Dynamic Weather-Impacted Environment,' *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, pp. 3256–3271, May 2017. DOI: 10.1109/TSMC.2016.2573271.

[148] C. P. Padhy, D. Sen and P. K. Bhaskaran, 'Application of wave model for weather routing of ships in the north indian ocean,' *Natural Hazards*, vol. 44, pp. 373–385, May 2008. DOI: 10.1007/s11069-007-9126-1.

[149] K. Takashima, B. Mezaoui and R. Shoji, 'On the fuel saving operation for coastal merchant ships using weather routing,' *Marine Navigation and Safety of Sea Transportation*, pp. 457–462, 2009.

[150] G. Mannarini, G. Coppini, P. Oddo and N. Pinardi, ‘A prototype of ship routing decision support system for an operational oceanographic service,’ *TransNav : International Journal on Marine Navigation and Safety of Sea Transportation*, vol. Vol. 7, pp. 53–59, Mar. 2013. DOI: `10.12716/1001.07.01.06`.

[151] J. Szlapczynska, ‘Multi-objective weather routing with customised criteria and constraints,’ *Journal of Navigation*, vol. 68, pp. 338–354, May 2015. DOI: `10.1017/S0373463314000691`.

[152] M. Grifoll, C. Borén and M. Castells-Sanabra, ‘A comprehensive ship weather routing system using CMEMS products and A* algorithm,’ *Ocean Engineering*, vol. 255, p. 111 427, 2022.

[153] K. Kepaptsoglou, G. Fountas and M. G. Karlaftis, ‘Weather impact on containership routing in closed seas: A chance-constraint optimization approach,’ *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 139–155, May 2015. DOI: `10.1016/j.trc.2015.01.027`.

[154] H.-B. Wang, X.-G. Li, P.-F. Li, E. I. Veremey and M. V. Sotnikova, ‘Application of real-coded genetic algorithm in ship weather routing,’ *The Journal of Navigation*, vol. 71, no. 4, pp. 989–1010, 2018.

[155] P. Krata and J. Szlapczynska, ‘Ship weather routing optimization with dynamic constraints based on reliable synchronous roll prediction,’ *Ocean Engineering*, vol. 150, pp. 124–137, May 2018. DOI: `10.1016/j.oceaneng.2017.12.049`.

[156] Copernicus, *Global ocean 1/12 physics analysis and forecast updated daily product. eu copernicus marine service information [data set]*, 2019.

[157] H. L. Tolman *et al.*, ‘User manual and system documentation of WAVEWATCH III TM version 3.14,’ *Technical note, MMAB Contribution*, vol. 276, no. 220, 2009.

[158] X. Lang, ‘Development of speed-power performance models for ship voyage optimization,’ *THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING*, 2020.

[159] B. Barras, ‘Ship design and performance for masters and mates,’ *Oxford: Elsevier*, 2004.

[160] H. N. Psaraftis and C. A. Kontovas, ‘Speed models for energy-efficient maritime transportation: A taxonomy and survey,’ *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 331–351, Jan. 2013. DOI: `10.1016/J.TRC.2012.09.012`.

[161] IMO, ‘Fourth imo ghg study 2020 - doc. mepc59/inf.10,’ *International Maritime Organization (IMO),London, UK*, 2020.

[162] H. Hoan, ‘Sfoc optimisation methods for man b&w two-stroke imo tier ii engines,’ *MAN Diesel & Turbo*, 2012.

[163] S. A. Harvald, 'Resistance and Propulsion of Ships,' p. 353, 1992.

[164] H. van den Boom, H. Huisman and F. Mennen, 'New guidelines for speed-power trials,' *SWZ Maritime*, p. 11, 2013.

[165] J. Holtrop, G. M. .-. I. S. Progress and undefined 1982, 'An approximate power prediction method,' *academia.edu*, 1982.

[166] S. Liu, B. Shang, A. Papanikolaou and V. Bolbot, 'Improved formula for estimating added resistance of ships in engineering applications,' *Journal of Marine Science and Application 2016 15:4*, vol. 15, pp. 442–451, Oct. 2016. DOI: 10.1007/S11804-016-1377-3.

[167] N. Bialystocki and D. Konovessis, 'On the estimation of ship's fuel consumption and speed curve: A statistical approach,' *Journal of Ocean Engineering and Science*, vol. 1, pp. 157–166, May 2016. DOI: 10.1016/j.joes.2016.02.001.

[168] E. Zermelo, 'Uber das Navigationsproblem bei ruhender oder veranderlicher Windverteilung,' *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 11, no. 2, pp. 114–124, 1931. DOI: https://doi.org/10.1002/zamm.19310110205. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/zamm.19310110205.

[169] I. Uber Technologies, 'H3 - Hexagonal Hierarchical Geospatial Indexing System,' *Software*, 2018.

[170] E. W. Dijkstra *et al.*, 'A note on two problems in connexion with graphs,' *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[171] P. E. Hart, N. J. Nilsson and B. Raphael, 'A formal basis for the heuristic determination of minimum cost paths,' *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100–107, 1968. DOI: 10.1109/TSSC.1968.300136.

[172] R. Ebendt and R. Drechsler, 'Weighted a* search - unifying view and application,' *Artificial Intelligence*, vol. 173, pp. 1310–1342, Sep. 2009. DOI: 10.1016/J.ARTINT.2009.06.004.

[173] C. Frasinaru and M. Raschip, 'Greedy best-first search for the optimal-size sorting network problem,' *Procedia Computer Science*, vol. 159, pp. 447–454, Jan. 2019. DOI: 10.1016/J.PROCS.2019.09.199.

[174] D. E. Goldberg, *Genetic algorithms*. pearson education India, 2013.

[175] S. Mirjalili and S. Mirjalili, 'Genetic algorithm,' *Evolutionary Algorithms and Neural Networks: Theory and Applications*, pp. 43–55, 2019.

[176] J. Bradbury *et al.*, *JAX: Composable transformations of Python+NumPy programs*, version 0.3.13, 2018.

[177] D. Shankar, P. Vinayachandran and A. Unnikrishnan, 'The monsoon currents in the north indian ocean,' *Progress in Oceanography*, vol. 52, no. 1, pp. 63–120, 2002. DOI: 10.1016/S0079-6611(02)00024-1.

[178] J. Blackledge, E. Coyle, D. Kearney and R. McGuirk, ‘Analysis of Wind Velocity Signals for Estimating the Wave Power Density of Ireland’s Coastal Resource,’ Jan. 2012. DOI: `10.1049/ic.2012.0176`.

[179] J. C. Butcher, ‘A history of Runge-Kutta methods,’ *Applied numerical mathematics*, vol. 20, no. 3, pp. 247–260, 1996.

[180] J. Santiago *et al.*, ‘Index of abundance of juvenile bigeye tuna in the Atlantic Ocean derived from echosounder buoys (2010-2020),’ *Collect. Vol. Sci. Pap. ICCAT*, vol. 78, no. 2, pp. 231–252, 2021.

[181] M. N. Maunder, H. Xu, C. Minte-Vera, J. L. Valero, C. E. Lennert-Cody and A. Aires-da-Silva, ‘Skipjack tuna in the Eastern Pacific Ocean, 2021: Interim assessment,’ 2022.

[182] M. Jeon, Y. Noh, Y. Shin, O. Lim, I. Lee and D. Cho, ‘Prediction of ship fuel consumption by using an artificial neural network,’ *Journal of Mechanical Science and Technology*, vol. 32, pp. 5785–5796, 2018.

[183] T. Uyanık, Ç. Karatuğ and Y. Arslanoğlu, ‘Machine learning approach to ship fuel consumption: A case of container vessel,’ *Transportation Research Part D: Transport and Environment*, vol. 84, p. 102 389, 2020. DOI: `https://doi.org/10.1016/j.trd.2020.102389`.

[184] W. Tarelko and K. Rudzki, ‘Applying artificial neural networks for modelling ship speed and fuel consumption,’ *Neural Computing and Applications*, vol. 32, no. 23, pp. 17 379–17 395, 2020.

[185] K. Bedi, D. Gómez-Ullate, A. Izquierdo and T. F. Montblanc, ‘Validating Ocean General Circulation Models via Lagrangian Particle Simulation and Data from Drifting Buoys,’ in *Computational Science – ICCS 2019*, J. M. F. Rodrigues *et al.*, Eds., Cham: Springer International Publishing, 2019, pp. 253–264.

[186] R. Vettor, M. Tadros, M. Ventura and C. Guedes Soares, ‘Route planning of a fishing vessel in coastal waters with fuel consumption restraint,’ *Maritime Technology and Engineering*, vol. 3, pp. 167–173, 2016.

[187] I. Granado *et al.*, ‘Towards a framework for fishing route optimization decision support systems: Review of the state-of-the-art and challenges,’ *Journal of Cleaner Production*, vol. 320, p. 128 661, 2021.