

Predicting CO₂ Emissions from a Fleet of Oil Tankers using Orange Data Mining

Jorge Luiz do Carmo¹

¹Vessel Operations Executive, Shipping Management, at Petrobras Petróleo Brasileiro, Rio de Janeiro, Brazil.

E-mail: jorgedocarmo58@yahoo.com.br

Abstract:

Purpose:

This study develops machine learning (ML) models using Orange Data Mining to predict CO₂ emissions from oil tankers, offering a scalable and user-friendly approach to emissions monitoring.

Methods:

Two datasets were analyzed: one combining technical specifications with voyage data, and another using voyage data alone. ML models were trained, optimized through hyperparameter tuning, and validated using cross-validation. Their performance was benchmarked against traditional bottom-up estimation methods.

Key Findings:

- AdaBoost yielded the best results:
 - Technical + Voyage data: MAE = 19.66, MAPE = 8.74%, R² = 0.991
 - Voyage-only data: MAE = 21.78, MAPE = 11.23%, R² = 0.985
- All ML models significantly outperformed the bottom-up approach (MAE = 102.6).

Conclusion: The results demonstrate that Orange Data Mining offers an accurate, no-code solution for CO₂ emissions prediction, supporting the maritime industry's decarbonization goals and regulatory compliance.

Keywords: CO₂ emissions prediction, maritime decarbonization, machine learning in shipping, orange data mining, emission monitoring systems, sustainable shipping, big data in maritime transport

Date of Submission: 03-06-2025

Date of Acceptance: 13-06-2025

I. Introduction

The success of the international economic system is highly dependent on the efficiency of maritime transport. However, ships are an increasing source of pollution, and the shipping sector is a substantial source of a variety of greenhouse gases (GHG) and other types of pollutants as well. The worldwide shipping community has been devoting increased attention, because of the gradual warming of the global climate, to the problem of GHG emissions, especially carbon dioxide (CO₂), which need to be drastically reduced to avert the most severe effects of climate change. Currently, the marine sector is struggling to overcome both new market obstacles and new regulatory challenges. Considering the seriousness of the challenge provided by climate change and the considerable amount of focus that has been placed on this topic, the International Maritime Organization (IMO) and the European Union (EU) have been exerting a lot of effort toward the limitation and reduction of GHG emissions from international shipping. Both have addressed the problem of vessel pollution and have required a gradual decrease in the air emissions that are generated by using marine fuels.

The concept of big data has quickly become the buzz of the industry, and the marine sector should expect both new possibilities and new issues brought about by it. The maritime sector creates enormous volumes of data, both through the operations of ships and from external sources. New environmental restrictions and growing competition are pressuring the industry to find solutions to optimization issues. This is an issue that may be able to be remedied, at least in part, via the use of data analytics since the models that are created from analytics may assist decision makers in being more effective and efficient.

Data mining is a method that allows for exploration and analysis by identifying significant patterns in a huge quantity of data via the use of algorithms. Orange Data Mining is a powerful open-source tool that enables users to explore and visualize data in a quick and easy manner, through a broad variety of data exploration, modelling, and visualization tools.

Predicting ships' CO₂ emissions is not only a strategic but also a necessary challenge for shipping companies due to the comprehensive and ever-increasingly rigorous air pollution programs. The ability to make accurate forecasts of CO₂ emissions is of critical importance for determining the most effective strategies for cutting those emissions. Studies have been done on emissions caused by marine transportation and provide a

variety of approaches that may be used to establish estimates or construct emission inventories. Using machine learning (ML) models to predict response values to a given collection of predictor data has shown to be a reliable and accurate approach with applications in the shipping industry.

II. Literature Review

2.1. The Importance of Predicting CO₂ Emissions in Shipping

Shipping is a highly controlled industry of the economy. Both the EU and the IMO have set considerable targets in terms of cutting GHG emissions from ships. As part of the EU's climate and energy framework for the year 2030, the EU adopted regulations to cut emissions by at least 40% by the year 2030. The maritime industry is responsible for around 3–4% of the EU's total CO₂ emissions, making it a significant contributor to the problem [1]. In 2011, IMO established the first worldwide regulations to increase ships' energy efficiency. In the last ten years, IMO has continued to act, including the approval of further regulatory measures. To facilitate their implementation, IMO has been carrying out a comprehensive program of capacity development and technical assistance, which includes a wide variety of worldwide initiatives [2].

The most important GHG released by ships is CO₂. IMO GHG strategy envisions a reduction in the carbon intensity of international shipping (to reduce CO₂ emissions per transport work) of at least 40% by the year 2030, with the goal of reaching 70% by the year 2050. This reduction is in comparison to the level of carbon intensity in international shipping in 2008 [2].

The primary concern of the shipping industry for ship energy efficiency management and pollution gas emission control is the prediction of emissions and ship energy consumption. Due to the increase in the volume of global shipping trade, they are attracting more global attention and research interest. A key factor in the development of the shipping industry's low carbon is the ability to accurately predict CO₂ emissions from maritime fleets [3]. Predicting the CO₂ emissions from ships has become not only a strategic but also a necessary duty for shipping industry because of the wide and more stringent regulations on air pollution [4].

2.2. Big Data in Shipping

It is a well-established fact that having access to a larger amount of data, in particular historical data, may often result in improved accuracy in model predictions [5]. Even though the shipping industry has access to a higher amount of data than it ever has before, mostly because of developments in sensor and networking technologies and capabilities, it is still struggling to unleash its full potential. These difficulties are mostly brought on by the fact of ships produce a significant amount of data from many sources and in various forms [6].

To identify hidden patterns and trends, big data analysis looks for connections between many measurable or immeasurable characteristics. This analysis will have a substantial influence on monitoring vessel performance and provide the ship operator performance forecasts, real-time transparency, and decision-making assistance.

For the marine sector, big data will likewise provide new possibilities and difficulties [7]. Big data technologies have been largely recognized as a breakthrough within the shipping industry, one that will alter most shipping operation patterns over the next one or two decades. This innovation has been widely recognized as a significant advancement within the shipping industry [8].

Big data analytics (BDA) has the potential to help the shipping and marine sector find solutions to a wide variety of difficult problems, such as estimating emissions [6]. Using a combination of artificial intelligence and machine learning-driven tools, algorithms, and processing systems, BDA is used to understand vast volumes of data. In terms of predicting CO₂ emissions, the concept behind BDA is to collect data pertinent to the issue and then use machine learning-based techniques to build a model that fits historical data better.

2.3. Maritime Data Sources and Uncertainties

Data of various kinds and scales of measurement (nominal, ordinal, interval, ratio) have historically been gathered in shipping, either manually or automatically using sensor technology.

It is possible to estimate the emissions caused by marine traffic using a variety of methodologies, each of which draws on a unique set of information resources. In the literature, to predict emissions plenty of variables are used related to ships technical specifications, voyages data, and weather data. Technical specifications include ship's type, length overall (LOA), gross tonnage (GT), design draft, deadweight (DWT), design speed, installed power of main and auxiliary engines, and design rotations per minute (RPM). Voyage data include information such as sailed distance, sailed time, average speed, fuel consumption, total weight of cargo loaded, and position of the ship. Weather data refers to information about winds, waves, ocean currents, temperature, and precipitations.

Technical ship specifications can be collected from commercial databases like IHS Fairplay or Clarksons Research. Voyage data is mainly informed by *noon reports* or Automatic Identification Systems (AIS), but it can be automatically collected through *internet of things* devices as well. Depending on the degree of detail desired,

one can access weather forecasts and historical data for free or for a fee. Some weather data also can be informed in the *noon reports* [9].

Empirical formulas widely spread in the literature can be applied to the technical databases for calculation of the ship main dimensions. However, technical data quality, data source accuracy, and update frequency are some potential uncertainties associated with ship characteristics [10].

AIS allows automatically identifying ships, primarily used to increase communication between ships, ship reporting, and navigational safety [11]. It is intended to have the ability to automatically transmit the ship's identity, type, location, speed, course, navigational status, and other safety-related information to other ships and coastal authorities [12]. Recently, AIS has received a lot of attention from the academic community and is widely used in predictions of shipping emissions. Because ships transmit data at intervals of a few seconds, the volume of AIS data is incredibly huge, presenting a practical challenge in its application. There are numerous problems in AIS data that must be considered prior to using the data [11]. Even though AIS offers maritime practitioners and academics a robust database that is simple to access, there is still a possibility that AIS data contains mistakes and inaccurate information. The data that is manually input into the system is where most of the mistakes and inaccuracies occur [8]. The information from the AIS may be noisy and unreliable, both too irrelevant and insufficiently relevant. On the other hand, it may be helpful for a variety of purposes when utilized properly [11].

The *noon report* is a ship voyage report data document prepared daily. Numerous aspects of the ship's nautical behavior are recorded, including ship location, distance travelled since the last report, average propeller revolutions, engine speed, sailing speed, cargo onboard, sea and weather conditions [13]. The *noon reports* continue to be the most common and widely used method in the maritime industry for monitoring the performance of individual ships [14]. On board and ashore, there are a range of management procedures that make use of the reports that are generated at noon. There is no universally accepted format for the *noon reports* that are maintained by the crew of a ship. Nonetheless, most operators gather data that is remarkably similar, and in most instances, the reporting frequency is the same (every day at noon while the ship is at sea). Besides, data from the *noon report* has inherent errors since the quality of measurement on board ships varies based on the procedures used [10]. Environmental conditions, such as the Beaufort number, are associated with an elevated level of aleatory error if they are not obtained from sensors. This aleatory error is made worse by the low resolution of the Beaufort scale, which results in rounding errors when converting from wind speed [9]. These reports are often created manually, which renders them prone to data entry mistakes [15]. If the completion of the report is not automated, any measurement involving the operation, reading, or recording of sensor values is susceptible to human error [26].

For more accurate carbon estimates, many studies focus on combining data from diverse sources, such as *noon reports*, AIS, ship sail weather data, and sensor data.

2.4. Machine Learning in Shipping

ML is a subfield of BDA that includes the application of mathematical or computational approaches to massive amounts of data to discover meaningful insights and patterns [16].

The availability of substantial quantities of data is a prerequisite for data-based innovations that make use of ML techniques. Historically, the maritime industry has faced significant challenges with large-scale data transmission and insufficient connectivity [17]. Over the last decade, ML learning has advanced dramatically and, as a result, these techniques are now useful in a far wider range of fields. Due to the abundance of data, ML has become a potent technique that has been utilized to effectively extract knowledge and complex patterns that may result in appealing ship energy reductions [18]. Mathematical correlations with relevant variables, such as operating speed, weather, and maintenance situation, may be used to represent a ship's performance. Extensive data from experiments or simulations may be used to build these relationships upon empirical equations, and ML can provide such equations [19]. Machine learning-based models are better suited for simulating high-dimensional situations due to their specialized fuel consumption models' biggest and toughest data needs. These models are generally used to assess the link between fuel consumption and characteristics, such as weather, ship load, wind and waves at sea, and operation policy, which cannot be described by principles [20].

Regression continues to be the most popular predictive modeling technique even in the big data age for a variety of reasons. First, regression's result is simple to understand. Second, it is possible to utilize the findings to identify the main components of the model. Third, the approaches may be applied to problems of almost any scale since they are parallelized, or at least they should be [5]. Linear regression (LR), multiple regression, ridge regression, and LASSO regression are the common ones used to predict fuel consumption and emissions.

In literature, supervised learning is mostly used to predict fuel consumption and emissions. Artificial Neural Network (ANN), Random Forest (RF), Support Vector Machines (SVM), and K-Nearest neighbor (KNN) are the common algorithms used.

There are ML libraries for the most prominent programming languages, such as Python, C++, R, Julia, Scala, etc. Python is the most popular programming language for ML and Data Science [21], and it is the language of choice of researchers.

III. Methods for Estimating Ships Emissions

In the vast literature, ships' CO₂ emissions are predicted using a variety of techniques, such as mathematical models, ML algorithms, and emission factors. In general, the approach used to forecast ships' CO₂ emissions relies on the data at hand, the accuracy required, and the resources available for analysis.

Mathematical empirical models include the top-down approach (fuel-based) and the bottom-up approach (activity-based). They are the two major ways often used to create ship emission inventories, based on multiple approaches that try to account for local, regional, and worldwide fuel consumption in shipping. The bottom-up techniques are the most often used methods for estimating emissions and thereby assessing compliance with emissions rules. Bottom-up techniques are primarily reliant on average figures for particular fuel consumptions and engine load factors [22]. The bottom-up approach is suggested when accurate sailing data are available. This method also requires several input parameters, including particular ship technical information (such ship types, engine specifications, and design specifics). For each specific ship activity, a quantity of emissions is scaled up across activities and voyages to get the total volume of emissions. Equation 1 is used to calculate the ships' emissions using a bottom-up approach.

$$E = P \cdot LF \cdot EF \cdot T \tag{1}$$

Where: E represents the total emissions, in grams of CO₂; P represents the engine power, in kW; LF represents the load factor; EF represents the emission factor for CO₂, in mass emitted per work output of the main engine in cruising, in g/kWh; and T represents the time in sailing, in hours.

The load factor is a measurement that indicates what percentage of the full power output of the engine is being used. The "propeller's law" has been utilized to provide support for a well-known assumption regarding the load factor, stating that utilization rate of the engine is equal to sailing speed over design speed to the power of 3 ("cubic rule"). The equation 2 is used to calculate the load factor.

$$LF = \left(\frac{V_s}{V_d}\right)^3 \tag{2}$$

Where V_s represents the sailing speed and V_d represents the design speed.

Ships engines are classified by IMO in slow-speed diesel engines (SSD), medium-speed diesel engines (MSD), and high-speed diesel engines (HSD) based on their speed ranges, as shown in table 1.

Table 1. Classification of engines based on speed ranges [23]

Classification	Criteria
SSD	Engines whose speed is 300 RPM or less
MSD	Engines whose speed ranges between 300 and 900 RPM
HSD	Engines whose speeds exceeds 900 RPM

MSD engines are used in various types of vessels, including cargo ships, ferries, and offshore support vessels. MSD engines are known for their efficient fuel consumption, durability, and versatility. They often use HFO or marine diesel oil (MDO) as their primary fuel. SSD are mainly used in large ships such as bulk carriers, tankers, and container vessels. SSD engines are characterized by their size, power, and efficiency. They are often powered by HFO. HSD engines are high-speed engines typically found in smaller vessels, such as yachts, workboats, and fast ferries. HSD engines are known for their compact size, lightweight design, and high power-to-weight ratio. They are commonly fueled by MDO or MGO.

The emission factor can be calculated by equation 3.

$$EF_f = \frac{EF_e}{SFC_{base}} \tag{3}$$

Where: EF_f represents the fuel-based emission (g CO₂/g HFO); EF_e represents the energy-based emission factor (g CO₂/kWh); and SFC_{base} represents the specific fuel consumption (g HFO/kWh). SFC_{base} may assume three distinct values, based on ship's generation, as per table 2.

Table 2. Proposed SFC (g/kWh) [23]

Type	Before 1983	1984-2000	2001+
SSD	205	185	175
MSD	215	195	185

Emissions per ship's voyage are calculated using equations 1, 2, and 3, with data from technical and voyages databases.

The bottom-up approach can be used with ML to predict ship emissions. Algorithms can be trained on historical emissions data and engine operating parameters to predict emissions for individual ships. For example, algorithms can be trained to identify patterns and relationships between fuel consumption rate, engine power, and emissions. Once the model is trained, it can be used to predict emissions for new ships based on their engine characteristics and operational parameters.

White-box Model (WBM), Black-box Model (BBM), and Grey-box Model (GBM) are methods commonly used to predict ship's fuel consumption, but they also can be used to predict ships' emission. The fundamental model construction procedure of the WBM involves calculating the resistances encountered by a ship from various sources using physical principles and hydrodynamics laws. The engine power and fuel consumption results from WBM can then be used to calculate the ship's emissions [24]. BBMs are classified into two categories: statistical models and ML models. In statistical models, probability is employed to infer the association between variables and fuel consumption. Applying ML techniques, on the other hand, provides accurate fuel consumption predictions using approximation functions [20]. WBM and BBM are combined to form GBM. ML may be employed in a WBM to enhance the physical models' accuracy in describing the basic processes that cause emissions. To increase the accuracy of the physical model, connections between engine operating parameters and emissions might be found using ML methods, for instance. In GBM, ML can be used to supplement physical models with empirical data. In a BBM, algorithms can be trained on historical emissions data and other relevant data such as engine operating parameters and environmental factors to identify patterns and relationships between the input variables and emissions.

To produce predictions, ML can be employed to discover patterns and correlations in the data. ML algorithms may be used to forecast CO₂ emissions by inputting ship data such as speed, cargo, and meteorological conditions. When analyzing complicated data sets, these models may be more accurate than mathematical ones.

BBMs algorithms include ANN, SVM, RF, KNN, LR, Gaussian Process (GP), and boosting algorithms.

IV. Orange Data Mining Tool

Data mining is the process of analyzing databases and identifying patterns in data, by gathering, cleaning, processing, analyzing, and drawing conclusions from them. Data mining on any significant scale necessitates the use of dedicated computer programs. Companies have developed tools and software to perform data mining procedures and acquire knowledge. Although these devices are useful, they may be quite expensive, and their high maintenance costs can build up rapidly. Thankfully, the field of free and open-source software is advancing rapidly. Numerous tools are now sufficiently developed to serve as viable alternatives to costly enterprises software.

Orange is an open-source, cross-platform data mining and ML software developed by the Bioinformatics Lab at the University of Ljubljana, Slovenia, together with the help of the open-source community, for explorative data analysis. Traditional programming languages, like Python and R, are powerful and adaptable, enabling precise control over the data analysis process. Nevertheless, to utilize them properly, users need to know sophisticated programming. Conversely, Orange is a non-code tool. It uses visual programming, a way that allows users to easily combine data analysis and interactive visualization techniques into powerful workflows. The components within Orange workflows that are responsible for reading, processing, and visualizing data are called widgets. Orange consists of a set of widgets for data pre-processing, features computer modeling, model comparison, and exploration methods. The widgets range from simple data visualization, subset selection and pre-processing, to empirical evaluation of learning algorithms and predictive modeling [25]. The widgets are integrated into processes using visual programming, and typically incorporate data processing or modeling methods, receiving input, and submitting the results as output. Orange widgets are represented by icons that have an input slot on the left and an output slot on the right, as shown in Figure 1.

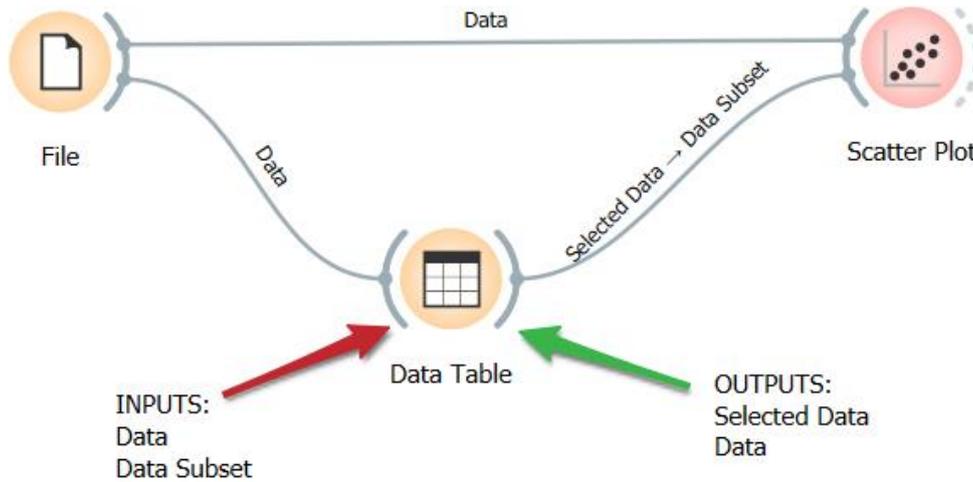


Figure 1. Widget input/output example [26]

The pipeline for processing data and information is defined by the widgets that are placed on the canvas and linked to their inputs and outputs. The workflow is handled instantly by the system. As soon as the widget gets the information, it handles it and sends the results.

V. Methodology

This study applies a quantitative research approach to develop and evaluate supervised machine learning models for predicting CO₂ emissions from a fleet of oil tankers, based on ship characteristics and voyage data. The methodology involved key steps including data preprocessing, algorithm selection, model building, hyperparameter tuning, performance evaluation, and comparison with mathematical (bottom-up) predictions. All stages were executed using Orange Data Mining (version 3.35). Computational efficiency was also emphasized, as it plays an important role in optimizing training time, model scalability, and handling large datasets effectively.

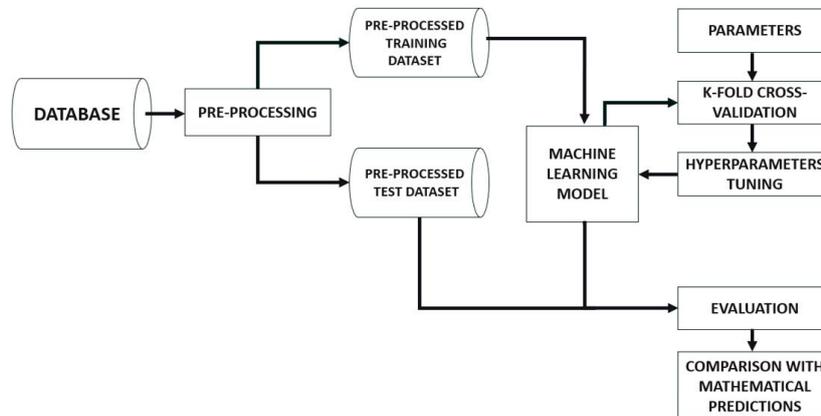


Figure 2. Proposed methodology

6.1. Preprocessing

Data preprocessing is a vital stage in any machine learning task, and arguably the most critical step in data mining. It involves transforming raw data into a format suitable for model training, ensuring that models learn from meaningful and consistent patterns. In this study, preprocessing included data acquisition, cleaning, feature selection, and exploratory analysis—phases that were entirely executed using Orange Data Mining, version 3.35.

Orange offers a robust set of tools that streamline the preprocessing pipeline through an intuitive, visual interface. Data acquisition was performed using Orange’s ability to import datasets from various sources such as Excel and CSV files, databases, or online repositories. Once imported, data were transformed into Orange’s internal data table format, allowing for seamless manipulation and analysis.

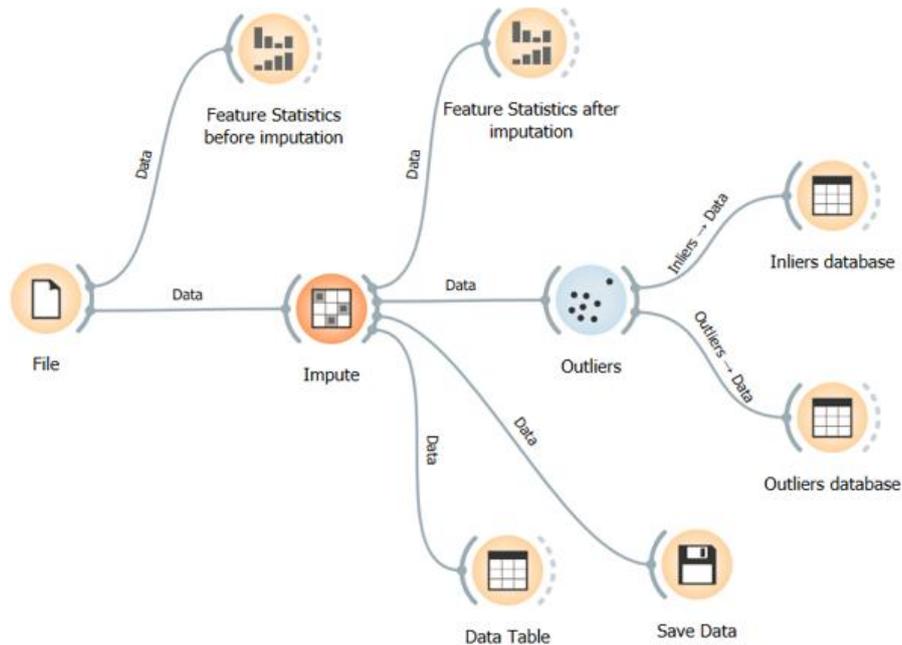


Figure 3. Workflow to display statistics, impute values, and identify outliers

High-quality data in sufficient quantity is essential for constructing reliable models. In this study, two databases were employed: a voyage database comprising twelve years of operational data from 292 oil tankers, and a technical database containing vessel specifications from IHS Fairplay. These were combined to create the “technical and voyage” database. Orange’s “Feature Statistics” widget was initially used to explore descriptive statistics, visualize distributions, and identify missing values and outliers.

The data cleaning phase began with the removal of voyages under 30 nautical miles and records with anomalous CO₂ values. Missing data were addressed using the “Impute” widget, specifically through a model-based imputer (simple tree), which builds a predictive model for each attribute based on the other available features. After imputation, outliers were identified using the “Outliers” widget, which supports multiple algorithms including One-class SVM, Covariance Estimator, Local Outlier Factor (LOF), and Isolation Forest. For this study, LOF was selected due to its efficiency in moderately high-dimensional data. The detected outliers were visualized using Orange’s “Scatter Plot” widget, which provides an interactive display to distinguish outlier values from normal data points.

Feature selection was carried out using Orange’s “Select Columns” widget. This tool enables users to manually define which variables should be used as predictors and which as the target. Features that lacked relevance or interpretability were excluded from further analysis. The target variable defined was CO₂ emissions.

Orange also facilitated data visualization through tools such as “Scatter Plot,” “Box Plot,” and “Feature Statistics,” enabling users to explore relationships between variables and identify trends, patterns, and anomalies. For instance, scatter plots revealed that VLCCs (Very Large Crude Carriers) showed the highest CO₂ emissions compared to other ship classes, confirming domain-specific expectations.

At the end of the preprocessing stage, two datasets were finalized for model training: Database 1 (technical and voyage data) and Database 2 (voyage data only). Each dataset contained 49,119 records, with Database 1 comprising 15 features and Database 2 comprising 8. These clean datasets formed the foundation for the subsequent modeling phase.

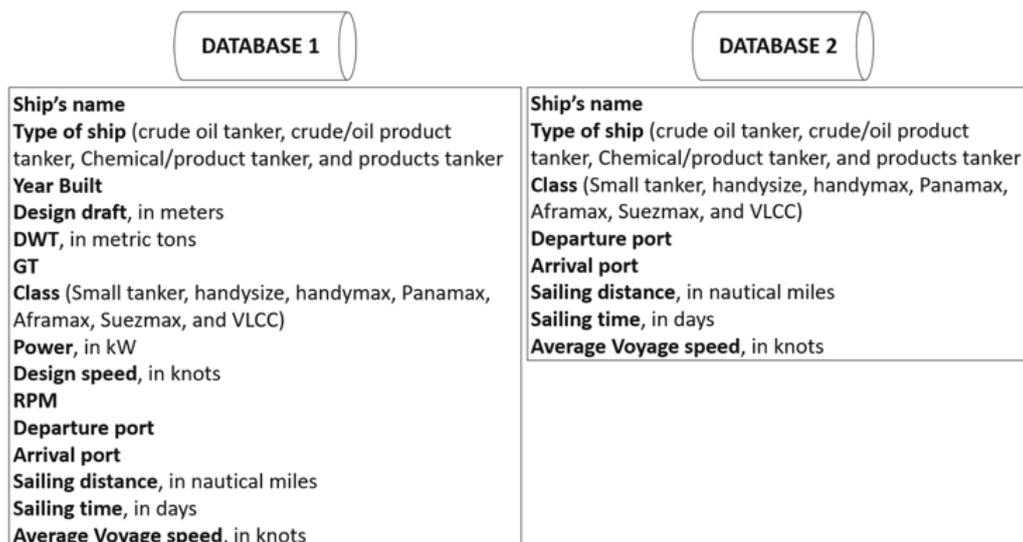


Figure 4. Features of each database

6.2. Models Selection

Choosing the right modeling methods is the key to making accurate predictions for different ship features and ways of operation.

BBMs were used to solve the complex, nonlinear relationship between CO₂ emissions and all other variables, considering their capacity to learn from historical ship data. In this study, the following algorithms were selected to build the models: Adaboost, Gradient Boosting (GB), KNN, RF, and LR.

6.3. Building the Models

Model development in this study was conducted entirely within the Orange Data Mining environment, which offers a visual and modular approach to machine learning. Orange allows users to build complete workflows by connecting widgets that represent each step of the modeling pipeline — from data preprocessing to algorithm training, evaluation, and prediction — without writing a single line of code. This design promotes transparency, flexibility, and ease of use, especially for complex tasks such as ensemble learning.

The study evaluated five supervised learning algorithms available in Orange: Adaboost, Gradient Boosting (GB), Random Forest (RF), K-Nearest Neighbors (KNN), and Linear Regression (LR). These algorithms were selected for their suitability in regression problems and their full integration into Orange's interface, which allows detailed configuration of parameters and evaluation strategies.

To ensure robust and unbiased model evaluation, 10-fold cross-validation was implemented using Orange's dedicated "Test and Score" widget. In this approach, the dataset is split into ten equally sized folds; each model is trained in nine folds and tested on the remaining one. This process is repeated ten times, with each fold serving as the test set once. The cross-validation framework provides a reliable estimate of model performance by reducing variance caused by data partitioning and preventing overfitting.

Each model was connected to the "Test and Score" widget, which automatically computes a set of performance metrics across all folds. These include Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Coefficient of Determination (R^2), and Relative Root Mean Square Error (RRMSE)—metrics appropriate for regression analysis. This widget also facilitates the comparison of multiple algorithms under identical conditions, enabling the selection of the most effective model.

Among the tested algorithms, Adaboost and Gradient Boosting (GB) are ensemble methods designed to improve performance by combining multiple weak learners. Adaboost builds its model sequentially by adjusting the weight of incorrectly predicted instances, thereby forcing subsequent learners to focus on harder cases. Orange allows for adjustment of key Adaboost parameters, including the number of estimators, learning rate, and loss function (linear, square, or exponential). Classification variants such as SAMME and SAMME.R are also supported.

Gradient Boosting was tested using Orange's implementations via Scikit-learn, CatBoost, and XGBoost. Unlike Adaboost, which modifies sample weights, GB corrects the residual errors of previous learners in each iteration, aiming to minimize the total loss function. The user can set the number of boosting iterations, learning rate, tree depth, regularization parameters, and random seed for reproducibility.

Random Forest (RF) aggregates predictions from multiple decision trees trained on bootstrap samples, introducing randomness in both data and feature selection to reduce variance. Orange enables fine control over

the number of trees, tree depth, number of attributes considered at each split, and options for class balancing and sample size thresholds.

K-Nearest Neighbors (KNN) is a non-parametric algorithm where prediction is based on the average outcome of the k most similar instances in the feature space. Orange’s KNN widget allows users to set the value of k, choose between distance metrics (Euclidean, Manhattan, Chebyshev, Mahalanobis), and define weighting schemes (uniform or distance-based), which influence the contribution of each neighbor to the final prediction.

Linear Regression (LR) served as the baseline model. Orange offers configuration for intercept fitting and regularization techniques such as Lasso (L1) and Ridge (L2), which help manage multicollinearity and prevent overfitting. The LR widget also allows users to control model complexity and interpret coefficients.

Once the models were configured, they were connected in Orange’s workflow to the “Predictions” widget for deployment. This widget integrates the trained models with test data, producing real-time predictions for further analysis. The entire modeling process—training, validation, and comparison—was executed visually, promoting clarity and reproducibility throughout the study.

6.4. Hyperparameters Tuning

Every learning algorithm comes with a set of hyperparameters attached to it. The iterative process of ML aims to discover the optimal combination of parameters that yield the best performing model. In most ML algorithms, there are certain parameters that influence the complexity of the model being fitted. These parameters cannot be directly estimated from the training data, and there is no analytical formula available to determine their optimal values. As a result, these parameters must be specified beforehand when fitting the predictive model or optimized through methods like cross-validation.

Grid Search, manual search, and random Search are techniques used for hyperparameter optimization in ML models. They differ in their approaches to exploring the hyperparameter space and finding the optimal set of hyperparameters. The most common ways to optimize hyperparameters are grid search and manual search. Manual tuning refers to a strategy in which a practitioner chooses hyperparameter settings based on their own personal knowledge as well as external influences, such as the findings of studies published in the relevant field of study. Grid search is the process of assessing the cartesian product of a limited set of values for each hyperparameter to identify optimal values for an ML model. This evaluation is done using a grid. Evaluation is performed on every conceivable combination of hyperparameter values that are a part of the subset that has been specified as the search space. Random search is a method in which hyperparameter values are taken at random (e.g., from a uniform distribution) from a given hyperparameter space.

The decision between grid and random layouts is determined by criteria such as the size of the hyperparameter space, available computing resources, and prior knowledge about the hyperparameters' relationship with model performance. Even though grid search and random search have different approaches, it is possible to use them together to efficiently search for the optimal set of hyperparameters. Combining them allows the exploration of the hyperparameter space first with random search and then conduct a more in-depth and targeted search with grid search.

A comparison between grid and random layouts is presented in Figure 5.

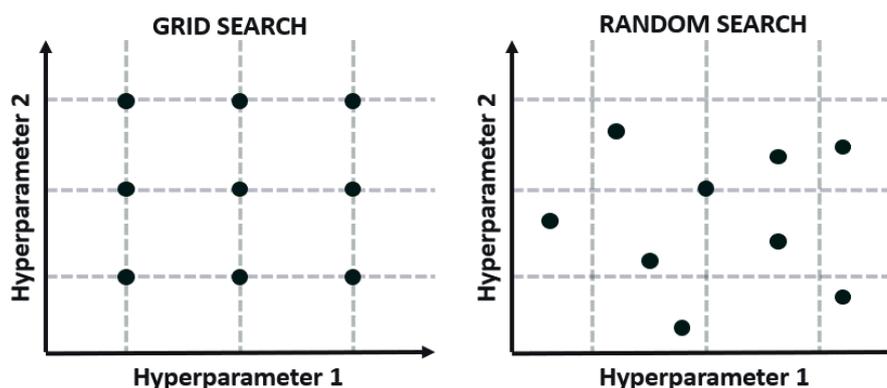


Figure 5. Comparison of grid search and random search layouts

The following steps were followed when both approaches were used together:

- Definition of the hyperparameter search space
- Random search
- Promising regions identification
- Refining the search space
- Grid search

- Evaluation and selection of the best hyperparameters

6.5. Evaluating the Models

To assess the performance of a ML algorithm on a given dataset, it is important to measure the level of agreement between the predictions made by the trained model and the actual observed values of the target variable. This involves quantifying the proximity between the predicted and observed values for each observation, with the aim of determining the degree of similarity between them.

When developing a model that predicts future outcomes using historical data, one of the most significant concerns is the possibility of overfitting. When a model has been overfit, it indicates that it has been trained to fit a particular set of training data extremely well, but it does not generalize well, which means that it does not perform well when attempting to predict the outcomes of observations that were not used for training.

Training the models allows them to be adjusted such that they reduce the loss functions as much as possible. It is expected that the model's performance will continue to improve with each adjustment. Nevertheless, there comes a point when making more adjustments to the model will no longer increase its overall performance. It is overfitting. Overfitting occurs when the model being used is excessively complicated in comparison to the quantity of training data and the level of noise in that data. Conversely, underfitting is the opposite of overfitting, and it happens mainly when the model is too simple to perform well with the training data or with the test data [27].

The only way to know how well a model will generalize to new instances is to test it out on new cases. This is the only method to determine how well a model will generalize. Putting the model into production and seeing how well it does there is one approach to achieving this goal. If the model, on the other hand, performs poorly, this is not a good plan. The database should be divided into two sets: the training set and the test set. This is the preferable course of action. The error rate that occurs when applying the model to new examples is referred to as the generalization error, and one may get an assessment of this error by evaluating the model using the test set. This number provides an indication of how well the model will function on examples that it has not seen before. It is a common practice to utilize 80% of the data for training while reserving 20% for testing.

The prediction potential of any algorithm may be improved by adjusting its hyperparameters, which are a set of parameters that need to be altered before training can begin. The goal of hyperparameter optimization is to reduce the amount of deviation that exists between the training data and the test data. A sample of the data that was not utilized to train the model is referred to as a validation dataset. This dataset is used to provide an estimate of the model's competence while the hyperparameters are being tuned.

The widget "Test and Score" permits evaluating the performance of a model by deploying it to a database and obtaining evaluation metrics. It allows connecting a trained model with test data and observing the model's predictions. The widget also can be used to compare different algorithms.

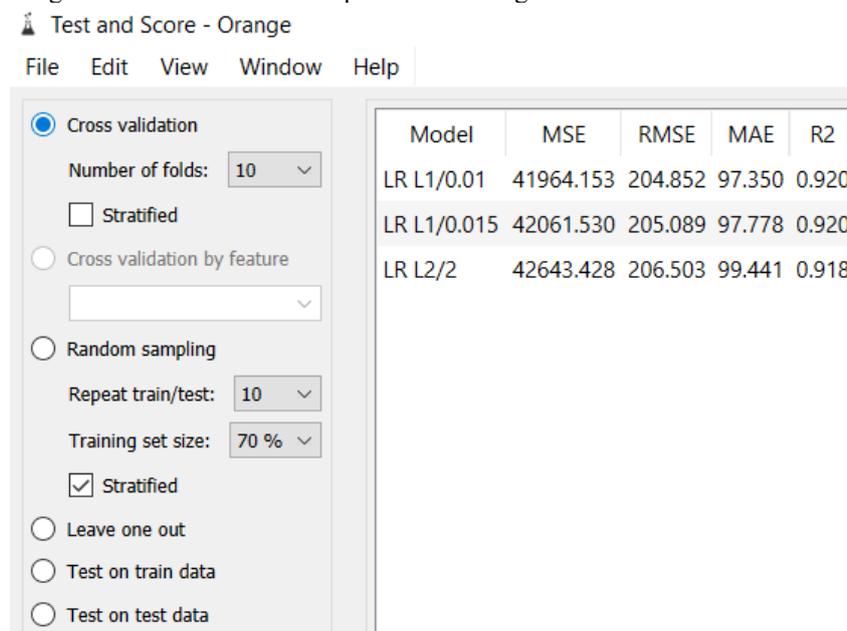


Figure 6. Widget "Test and Score"

The widget "Test and Score" has six options for sampling the models:

- Cross-validation
- Cross-validation by feature
- Random sampling
- Leave one out
- Test on training data
- Test on test data

Cross-validation is a resampling technique to determine the precision of a model for making predictions on a brand-new set of data. A training portion and a test portion of the data set are separated for this. The training part is split into k subsets, with k-1 representing the training data used to estimate the parameters and the remaining subsets used to assess performance. The approach is repeated to guarantee that each subset takes part in model training and validation. The division of the set makes sure that the model trains, lowers its error, and generalizes successfully. Cross-validation can be set with 2, 3, 5, 10, or 20 folds. The default value is 10 folds. Cross-validation may also be done feature-by-feature, with the folds chosen depending on the category feature selected from the meta-features.

Random sampling divides the data at random into the training and testing sets in the defined ratio, for example 70:30. The whole process is repeated a predetermined number of times.

An analogous method is leave-one-out, in which one instance is withheld for model training. As each instance is utilized as a part of the training set in several rounds, the fundamental benefit of the Leave-One-Out cross-validation approach is that it utilizes almost all the data that is available for training. This method is highly stable and reliable, but it is noticeably slower due to the process's exhaustive nature. When the dataset is small, this may result in a more accurate assessment of the model's performance than conventional cross-validation methods. However, since Leave-One-Out cross-validation involves training and testing the model for each instance separately, it may be computationally costly, especially for big datasets. Since each instance is utilized as a testing set separately, it might potentially be sensitive to outliers in the dataset.

For testing on the training data, the entire dataset is utilized for both training and testing. However, this strategy typically produces inaccurate outcomes.

It is possible to test on test data by incorporating a distinct dataset containing testing examples, which can be derived from a different file or specified within a different widget.

In this study, k-fold cross-validation was used, with k=10, to divide the training and test data into 10 parts of equal or close sizes. At each iteration, 9 parts were used in model training, with different configurations – the hyperparameters. The other remaining part was used for performance estimation. This means that cross-validation method splits the training dataset up into 10 subsets and then rotates them 10 times before applying them to the validation process, which results in an increase in the initial amount of data by a factor of 10. Figure 7 illustrates a 10-fold cross-validation.

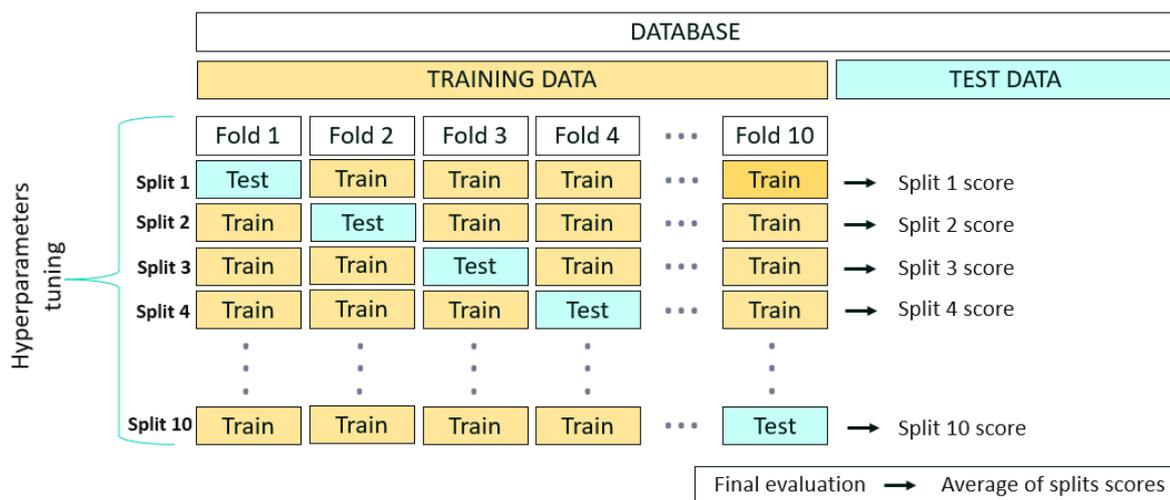


Figure 7. 10-fold cross-validation

The following metrics are provided by Orange, displayed in the widget “Test and Score”:

- Mean Square Error (MSE)
- Root Mean Square Error (RMSE)
- Mean Absolute Error (MAE)
- Coefficient of Determination (R²)

The metrics can be calculated by:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2 \quad (4)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2} \quad (5)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}| \quad (6)$$

$$R2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2} \quad (7)$$

Where \hat{y} is the predicted value of y , and \bar{y} is the mean value of y .

MAE measures the arithmetic mean of the deviations between actual and predicted values. The lower the MAE, the better the model's performance, as it indicates smaller errors between the predicted and actual values.

MSE is the most widely used measure in regression settings. MSE measures the errors between them, and RMSE is equal to the square root of the MSE. The RMSE and the MAE are both ways to measure how far apart the vector of forecasts and the vector of target values are [27]. MAE and RMSE are the easiest measures to manipulate because they are calculated on the variable of interest, in this study, in metric tons of CO₂ emitted. The smaller the values, the smaller the differences between the emissions predicted by the models in comparison with the observed emissions. The benefit of MAE is that it is unaffected by outliers. This measure just takes into consideration the magnitude of the individual errors, disregarding their sign.

R2 is the most common approach for measuring how well the model fits the data and correctly predicts. The value of the R2 indicates the entire percentage of the variation in the dependent variable that can be attributed to the independent variable. R2 varies between 0 and 1 and is an indication of how much the model explains the data. Any value greater than zero shows how much variability in the data the model can explain.

6.5. Constraints of Methodology

Despite Orange Data Mining providing a rich collection of algorithms, it's worth mentioning that the availability of specific advanced or specialized algorithms might be limited compared to other dedicated ML libraries or frameworks. Besides, Orange Data Mining has predefined algorithms with limitations on customization or flexibility in tuning the hyperparameters for developing complex models tailored to the specific CO₂ prediction problem.

The interpretability of the BBMs can be a constraint. They provide highly accurate predictions but lack interpretability, making it challenging to explain the underlying factors influencing CO₂ emissions.

Training and evaluating ML models on large datasets requires substantial processing power, memory, and storage resources. Ensuring access to appropriate computational resources becomes crucial for efficient model development. The specification of the computer used in this study is consistent with the devices currently used by companies. However, better results in tuning the models could have been obtained if a computer with even greater processing and memory capacity were used.

VI. Results and Discussion

This results section presents the outcomes of the research, consisting of primary data and derived data. The original results obtained by different ML models are presented in the following.

6.1. Results of Adaboost Models

To evaluate the performance of Adaboost models in predicting CO₂ emissions from oil tankers, an extensive hyperparameter tuning process was carried out using random search followed by grid search, both supported by 10-fold cross-validation. The objective was to identify optimal combinations of model parameters, particularly the number of estimators and learning rates that minimize prediction errors while maintaining reasonable computational efficiency.

Database 1: Technical and Voyage Data

For the first dataset, a random search was initially conducted over a broad hyperparameter space defined, with estimators ranging from 20 to 1,500 and learning rates from 0.0001 to 1.0. Performance was evaluated based on the Mean Absolute Error (MAE) metric using Orange's "Test and Score" widget with the SAMME classification algorithm and a linear loss function.

The random search results showed that lower learning rates combined with both small and large numbers of estimators yielded the lowest MAE values. However, models with a high number of estimators incurred substantially longer processing times, highlighting the trade-off between performance and computational cost. To refine the model and improve efficiency, a grid search was performed in the narrowed range of 50 to 200 estimators. Estimators in the 125–150 range paired with learning rates between 0.0005 and 0.001 provided strong predictive accuracy with significantly reduced processing time compared to models using 1,500 estimators. The best model for Database 1 used 125 estimators and a learning rate of 0.001, achieving a MAE of 19.66, R² of 0.991, and a processing time of 1 hour and 16 minutes. While a configuration with 1,500 estimators achieved comparable accuracy (MAE = 19.95), its processing time exceeded five hours, making it less practical for real-time applications.

Database 2: Voyage Data Only

A similar procedure was applied to Database 2. The search space was adjusted to range from 30 to 800 estimators, with the same interval for learning rates. After a preliminary random search, a grid search was conducted in the range of 100 to 400 estimators. The most promising results were concentrated around 200 to 300 estimators with a learning rate of 0.05.

The best model using Database 2 employed 250 estimators and a learning rate of 0.05, producing a MAE of 21.78, R² of 0.985, and requiring 2 hours and 40 minutes for training. While the performance was slightly inferior to the model trained with technical data, the results still demonstrated high predictive power.

6.2. Results of Gradient Boost Models

To identify the best-performing Gradient Boosting (GB) models, this study applied a two-phase hyperparameter optimization process involving random search followed by grid search, supported by 10-fold cross-validation. The experiments focused on three implementations of GB available in Orange: GB Scikit-learn, GB CatBoost, and GB XGBoost, each evaluated with both Database 1 (technical + voyage data) and Database 2 (voyage data only).

GB Scikit-learn

For GB Scikit-learn models, an initial random search was conducted over a broad space of hyperparameters. This phase revealed that models with a higher number of trees and shallower depth achieved lower MAEs. Based on these findings, a refined grid search was carried out, narrowing the search space to trees between 1,000 and 2,000, depth between 6 and 10, and a learning rate of 0.7. The grid search results showed improved performance despite longer processing times due to the model's iterative structure.

The best-performing configuration for Database 1 used 1,500 trees, results rate of 0.7, and a tree depth of 10, achieving an MAE of 23.82 and R² of 0.987 in 5 hours and 41 minutes.

For Database 2, a similar process was followed, with the search space adapted. The final grid search focused on tree depths of 30, learning rates of 0.05–0.1, and 700–800 trees. The best results were obtained with 800 trees, a learning rate of 0.1, and a depth of 30, yielding an MAE of 24.16 and R² of 0.982, with a training time of 7 hours and 19 minutes.

GB CatBoost

For GB CatBoost, the initial random search covered trees up to 5,000, learning rates between 0.001 and 1, and depths between 2 and 10. Regularization was controlled by the lambda parameter. The most promising results from the random search indicated that higher tree counts and mid-range depths led to lower MAEs.

The subsequent grid search focused on 5,000 trees, learning rate of 0.3, lambda of 0.5 or 1, and depths between 4 and 10. The best model for Database 1 was achieved with 5,000 trees, a depth of 7, and lambda of 0.5, obtaining an MAE of 35.20, R² of 0.983, and training time under 33 minutes.

For Database 2, the grid search explored tree counts between 1,000 and 3,000, fixed lambda = 1, learning rate = 0.1, and tree depth = 10. The optimal model used 2,750 trees, producing an MAE of 36.46, R² of 0.980, and training time of 23 minutes.

GB XGBoost

The GB XGBoost models were tested using an extensive hyperparameter space, with trees ranging from 70 to 8,000, learning rates from 0.001 to 1, and tree depth from 2 to 8. No subsampling was applied — all features were used at every tree level and split, and lambda was fixed at 1 to apply moderate L2 regularization. The random search results showed that smaller learning rates and tree depths between 6 and 8 were most effective.

The grid search that followed focused on 1,000–8,000 trees, depth of 6–8, and learning rate of 0.3. The best model for Database 1 used 8,000 trees, depth of 7, and achieved an MAE of 21.23, R² of 0.987, with training time of 3 hours and 45 minutes.

For Database 2, a refined grid search concentrated between 400 and 900 trees, depth 30, and learning rate 0.3. The best-performing configuration included 900 trees, reaching an MAE of 22.18, R² of 0.986, and requiring 1 hour and 12 minutes of processing time.

In all three GB implementations, Database 1 consistently outperformed Database 2, confirming the added value of technical attributes in improving CO₂ emission prediction accuracy. Among the GB variants, XGBoost produced the best results overall, followed by Scikit-learn and CatBoost. However, CatBoost stood out for its superior speed, reaching competitive performance in less than half the time required by other models.

6.3. Results of KNN Models

The performance of K-Nearest Neighbors (KNN) models in predicting CO₂ emissions was evaluated using a series of hyperparameter configurations, tested via random search and assessed through 10-fold cross-validation. The analysis was conducted separately for both databases used in this study: Database 1 (technical + voyage data) and Database 2 (voyage data only).

KNN Models – Database 1

For models trained with Database 1, the hyperparameter search included variation in the number of neighbors (k), choice of distance metric (Euclidean or Chebyshev), and weighting scheme (uniform or distance-based). A total of 59 different configurations were tested using random search.

When uniform weighting was applied, MAE values tended to increase with larger numbers of neighbors across both distance metrics. However, when predictions were weighted by distances, higher values of k resulted in lower MAE values, suggesting that weighting neighbors based on proximity improves predictive accuracy, especially in higher-k regions.

The best-performing KNN models with Database 1 used the Euclidean distance, distance-based weighting, and a neighbor count of 21 to 23. The top result achieved an MAE of 22.93, R² of 0.985, and required only 1 minute of processing time.

KNN Models – Database 2

For Database 2, the search space was expanded to include up to 100 neighbors, while maintaining the same metric and weighting options. A total of 60 configurations were tested through random search.

The best results were again observed when using Euclidean distance and distance-based weighting, this time with a smaller number of neighbors ranging from 11 to 14. The top configuration reached an MAE of 38.06, R² of 0.948, and executed in under 50 seconds.

6.4. Results of Random Forest Models

To determine the optimal configuration of Random Forest (RF) models for predicting CO₂ emissions, a combination of random search and grid search was employed, supported by 10-fold cross-validation. The process was applied independently to both datasets — Database 1 (technical and voyage data) and Database 2 (voyage data only).

RF Models – Database 1

For Database 1, the initial random search explored a wide hyperparameter space. Key parameters included the number of trees (from 5 to 1,650), the number of attributes considered at each split (from 1 to 10), the limit depth of the trees (unlimited or fixed at 4, 5, or 6), and the minimum subset size for splitting (various thresholds). Performance was assessed using MAE as the primary metric.

Analysis of the random search outcomes revealed that higher numbers of trees and using the default number of attributes per split (square root of the total number, approximately 3.87) led to lower MAEs. Optimal MAE values are associated with models using deeper trees, default attribute splits, and a minimum subset of 2.

Subsequently, a grid search was conducted by narrowing the parameter space to trees ranging from 1,000 to 2,000, using approximately 3.87 attributes per split, unrestricted tree depth, and a minimum subset of 2. The top-performing configuration used 1,700 trees, achieving an MAE of 29.11, R² of 0.985, and a training time of 8 hours and 11 minutes. Other configurations, with 1,500 and 2,000 trees yielded nearly identical results, confirming the model's stability.

RF Models – Database 2

The same tuning methodology was applied to Database 2, with the adjusted search space. The maximum number of trees was limited to 1,000 due to the smaller feature set. The most promising results again aligned with configurations using the square root of the number of attributes per split (≈ 2.82), no depth limitation, and minimum subset size of 3.

The best configuration used 1,000 trees, producing an MAE of 31.54, R² of 0.982, and a processing time of 7 hours and 48 minutes. Models with 700 and 900 trees showed nearly identical performance, indicating robustness across a range of tree counts.

6.5. Results of Linear Regression Models

To identify the best linear regression (LR) models, a random search was conducted using 32 hyperparameter sets for Database 1 and 46 for Database 2. Lasso (L1), Ridge (L2), and Elastic Net regressions were tested with varying regularization strengths (alpha).

For Database 1, the best result based on the lowest MAE—was achieved by Lasso regression with $\alpha = 0.0001$ (MAE = 95.01, R² = 0.919) in 28 minutes of processing time.

For Database 2, Ridge regression consistently delivered the best performance across several alpha values, with the lowest MAE of 94.99 and R² = 0.919 in 18 minutes of processing time.

6.6. Results of Mathematical Calculations

Mathematical predictions were made using the bottom-up approach for comparison with ML models performance. The emission factors were calculated using the equation 3, considering fuel-based emission of 3.114 gCO₂/gHFO proposed by IMO (2020), and the imputed SFC, as per Table 2.

The emissions per ship’s voyage were calculated using the equations 1 and 2.

Metrics MSE, RMSE, MAE, and R2 were calculated using the equations 4, 5, 6, and 7. The results are shown in Table 3.

Table 3. Results of mathematical predictions

Model	MSE	RMSE	MAE	R2
Bottom-up approach	78115	279	102.6	0.835

6.7. Derived Data

Validation metrics were used to evaluate the performance of ML models. In other words, they provided information on how the models performed when applied to unknown data, estimating the magnitude of errors.

MSE, RMSE, MAE, and R2 were calculated by Orange, as primary data. Relative Root Mean Square Error (RRMSE) and Mean Absolute Percentual Error (MAPE) have been chosen as additional metrics, as derived data, to provide additional perspectives on the performance of the models by considering the relative and proportional errors in the predictions compared to the actual values.

RRMSE was calculated by equation 8, using RMSE:

$$RRMSE = \frac{RMSE}{\frac{1}{N} \sum_{i=1}^N \hat{y}} \tag{8}$$

where \hat{y} is the predicted value of y .

RRMSE calculations of the best models are shown in Table 4.

Table 4. RRMSE

Model	RRMSE Database 1	RRMSE Database 2
Adaboost	19.82%	24.77%
GB scikit-learn	27.22%	27.07%
GB catboost	26.40%	28.88%
GB xgboost	24.92%	23.57%
KNN	25.25%	46.35%
RF	24.99%	26.88%
LR	57.80%	57.71%

The widget "Predictions" receives a dataset and one or more predictors and outputs the data and the predictions. It displays predictive model conclusions. The output of the widget is another dataset to which new meta-attributes are appended based on predictions. One can observe the result in a data table, as shown in Figure 8. The actual CO₂ emissions in the first column and the predictions of a KNN model in the second column. The widget displays predicted values determined by the trained model's learned patterns. To evaluate the performance of the model, one can compare the predicted values from the testing database against the actual values.

EMISSIONS (ACT)	kNN	Fold	NG DISTANCE (M)	SAILING TIME (DAY)	VELOCITY SPEED
133.591	121.398	1	428.00	1.3	13.7179
166.91	150.461	1	557.00	1.5625	14.8533
65.394	63.5194	1	368.00	1.5	10.2222
31.14	38.2251	1	185.00	0.677083	11.3846
70.065	64.4995	1	360.00	1.375	10.9091
62.28	66.9689	1	363.00	1.27083	11.9016

Figure 8. Predictions of a KNN model with Database 2, using the widget “Predictions”

With the predictions made by the models, MAPEs were calculated using equation 9.

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{9}$$

MAPE calculations of the best models running with both databases are shown in Table 5.

Table 5. MAPE

Model	MAPE Database 1	MAPE Database 2
Adaboost	8.74%	11.23%
GB scikit-learn	13.38%	12.18%
GB catboost	19.64%	19.39%
GB xgboost	11.32%	12.03%
KNN	11.76%	18.48%
RF	15.31%	16.01%
LR	66.10%	66.01%

6.8. Interpretation

Descriptive statistics are vital for understanding, summarizing, exploring, and communicating the key characteristics of a database. They provide a foundation for further analysis, support decision-making, and assist in quality assessment and data exploration. These statistical analyses provide information about central tendency, variability, and distribution of the data, helping better understanding of the nature of the data and identify possible outliers or patterns.

Table 6 shows the descriptive statistics of the database used in this study.

Table 6. Descriptive statistics of the database

Statistic	Value
Mean	356.30
Median	157.57
Mode	62.28
Standard Deviation	723.08
Minimum	1.93
Maximum	12,586.17
First quartile	78,47
Third quartile	337.28

The mean of CO₂ emissions is 356.30 metric tons. This measure represents the central tendency of the data, where all values are summed and divided by the total number of observations. The median is 157.57 metric tons. This value represents the midpoint of the data, where half of the values are above, and half are below. The mode is 62.28 metric tons. It indicates the most frequently occurring values of CO₂ emissions. The standard deviation is 723.08 metric tons. This measure represents the spread of the emissions around the mean. A higher standard deviation indicates greater variability of the data from the mean. The minimum value is 1.93, representing the smallest value observed. The maximum value is 12,586.17, representing the largest value observed. The first quartile, also known as the lower quartile, divides the data into the lower 25% of values. It is the median of the lower half of the dataset. In this case, the first quartile is 78.47 metric tons, indicating that 25% of the values are below this threshold. The third quartile, also known as the upper quartile, divides the data into the upper 25% of values. It is the median of the upper half of the dataset. In this case, the third quartile is 337.28 metric tons, suggesting that 75% of the values are below this threshold. When using regression models, they intend to make predictions for values of a continuous variable, so the difference between predicted and actual values is unavoidable. Analysis made by Willmott and Matsuura [28] indicates that MAE is considered the most suitable metric for measuring the average magnitude of errors. In contrast to RMSE, MAE provides a clear and unambiguous indication of the average error magnitude. Based on this analysis, it appears that all evaluations and comparisons of average model performance errors, across different dimensions, should be centered around the use of MAE. Opinions among researchers are generally aligned in stating that the determination of whether a MAE is considered good or not relies on the specific characteristics of the data. MAE and RMSE outputs are indicated in metric tons of CO₂.

The interpretation of MAE (Mean Absolute Error) involves assessing model accuracy by comparing error magnitudes across models and datasets. A lower MAE indicates better predictive performance. The best result — MAE of 19.66 metric tons — means the model's average prediction error is only 5.52% of the average CO₂ emissions (356.30 metric tons), suggesting high accuracy.

In Database 1, the most accurate models were Adaboost, followed by GB xgboost and KNN. In Database 2, Adaboost again had the lowest MAE, followed by GB xgboost and GB scikit-learn.

Models like Adaboost, GB xgboost and GB scikit-learn showed consistent performance across both datasets, while KNN performed notably worse on Database 2, indicating limited generalizability.

MSE and RMSE are standard metrics for evaluating regression models, with RMSE preferred for its interpretability in the same units as the target variable (metric tons of CO₂). Lower values indicate better predictive accuracy. For Database 1, Adaboost and Gradient Boosting (XGBoost) showed the lowest MSE and RMSE, indicating superior performance. For Database 2, Adaboost, XGBoost, and Random Forest outperformed other models. Linear Regression had the highest MSE and RMSE in both datasets, reflecting the least accurate predictions.

In terms of R², the results indicate that both Database 1 and Database 2 have well-fitted models. For Database 1, the R² values range from 0.919 to 0.991, suggesting that the models can explain approximately 91.9% to 99.1% of the variance in the data. Similarly, for Database 2, the R² values range from 0.919 to 0.986, indicating that the models can explain around 91.9% to 98.6% of the variance. These high R² values signify a strong relationship between the predictors and the target variable, indicating good predictive performance of the models on both databases.

MAPE is a metric that quantifies the percentage of error in relation to the actual values. To determine how accurate the estimated amounts were, compared to the actual numbers, it estimates the average of the absolute percentage errors for each item in a dataset. The dataset must have non-zero values for MAPE to be effective for evaluating large datasets. A simple measurement, a MAPE of 10% shows an average difference of 10% between the predicted value and the actual data, independent of the direction of the variance (positive or negative). Many researchers adopt the interpretation of typical MAPE values proposed by Lewis [29], as shown in Table 7.

Table 7. Interpretation of typical MAPE values [29]

MAPE	Interpretation
<10%	Highly accurate forecasting
10% < MAPE < 20%	Good forecasting
20% < MAPE < 50%	Reasonable forecasting
>50%	Inaccurate forecasting

As shown in Table 8, based on the MAPE classification, the Adaboost model presented a highly accurate prediction with Database 1. This means that the predictions of this model have a low average percentage error in relation to the actual data values and its performance is considered excellent. With Database 2, the Adaboost

model performed well, while not as accurate as with Database 1. The other models had good predictions, with a MAPE between 10% and 20% in both databases, with the exception of LR, which with MAPE above 50% had poor performances, demonstrating a high mean absolute percentage error in its predictions.

Table 8. Models’ performance considering MAPE classification

Models	Database 1	Database 2
Adaboost	Highly accurate forecasting	Good forecasting
GB scikit-learn	Good forecasting	Good forecasting
GB catboost	Good forecasting	Good forecasting
GB xgboost	Good forecasting	Good forecasting
RF	Good forecasting	Good forecasting
KNN	Good forecasting	Good forecasting
LR	Inaccurate forecasting	Inaccurate forecasting

RRMSE is a dimensionless and relative metric used to compare models in percentage terms, which provides a relative measure of the error compared to the average predicted value. The lower the percentage, the better the performance. Many researchers also adopted the interpretation of typical RRMSE values proposed by Jamieson, Porter, & Wilson [30], as shown in Table 9.

Table 9. Interpretation of typical RRMSE values

RRMSE	Interpretation
<10%	Excellent performance
10% < RRMSE < 20%	Good performance
20% < RRMSE < 30%	Fair performance
>30%	Poor performance

Considering the RRMSE classification, the Adaboost model was the only one that performed well with Database 1, however the model performed well with Database 2. GB scikit-learn, GB catboost, and RF models performed well with Database 1 and Database 2. KNN model had reasonable performance with Database 1 and poor performance with Database 2. LR performed poorly with both databases.

Table 40. Models’ performance considering RRMSE classification

Models	Database 1	Database 2
Adaboost	Good performance	Fair performance
GB scikit-learn	Fair performance	Fair performance
GB catboost	Fair performance	Fair performance
GB xgboost	Fair performance	Fair performance
RF	Fair performance	Fair performance
KNN	Fair performance	Poor performance
LR	Poor performance	Poor performance

Processing time is an important consideration when building machine learning models. The time required to train and evaluate models can have significant implications for the development process, scalability, and practicality of deploying the models. Analyzing the processing times, KNN model stands out for having an extremely low processing time in both databases, taking only 1 minute in Database 1 and 47 seconds in Database 2. The GB catboost and LR models also featured relatively short processing times, taking 32 minutes in Database 1 and 23 minutes in Database 2 for GB catboost, and just 28 minutes in Database 1 and 32 seconds in Database 2 for LR. The Adaboost and GB xgboost models had intermediate processing times, taking a few hours on both databases, which can be considered a moderate processing time, compared to other models. GB scikit-learn and RF models show longer processing times, taking several hours to complete training and testing.

Table 11 presents the comparison between mathematical calculations and the best ML models. Compared to the mathematical calculation, the models had better performance, except LR. The MAE of 102.6 metric tons calculated using the bottom-up approach represents approximately 28.80% of the average emissions, which can

be inferred that the MAE is in a relatively medium error range in relation to the magnitude of the values and the variability of the data. MAPE of 38.97% shows an average difference of 38.97% between the predicted value and the actual data.

Table 11. Comparison between mathematical calculations and best ML models

Model	MSE	RMSE	MAE	R2	MAPE
Adaboost	4985	71	19.66	0.991	8.74%
GB scikit-learn	9437	97	23.82	0.983	13.38%
GB catboost	8864	94	35.20	0.983	19.64%
GB xgboost	7267	85	21.22	0.986	11.32%
KNN	8035	90	22.93	0.985	11.76%
RF	7950	89	29.10	0.985	15.30%
LR	42390	206	95.01	0.919	66.10%
Bottom-up approach	78115	279	102.6	0.835	38.97%

6.9. Discussion

This study used a large database with historical data of 49,119 voyages of 292 ships. It is a well-known fact that when models have access to more data, especially historical data, they can often make more accurate predictions. In literature, most studies are conducted on a small number of ships, making the robustness of the methodologies questionable.

Information on fuel consumption and CO₂ emissions from ships is not publicly available. The voyage database was made up of bulletins sent by the ship's captains at the end of each of the 49,119 voyages, over 12 years, with exclusive and specific information on the fleet of tankers under study, which reduces the chance of errors, if compared to the numerous AIS information and several *noon reports* sent during a voyage. These voyage bulletins provide valuable insights into the characteristics and actual performance of the fleet in terms of emissions. The use of models with only voyages data, database 2, enabled good predictions without using technical information obtained from paid databases. The models built with both databases did not depend on information from the main sources of maritime transport, *noon reports* and AIS, and, therefore, did not use weather information, such as winds, waves, ocean currents, temperature, and precipitations, nor ship positioning.

Ship owners, charterers, technical operators, and other stakeholders can derive significant benefits from using Orange Data Mining to build CO₂ emissions prediction models. It simplifies data analysis, allows the evaluation of emission reduction strategies, assists in decision-making, enables the negotiation of sustainable contracts, and facilitates compliance with environmental regulations.

To accomplish the intended performance, ML models require tuning and optimization. Quick processing times enable rapid experimentation, allowing for the analysis of various model architectures, algorithms, and data pre-processing techniques. Faster iterations contribute to a more efficient model development procedure, enabling more effective iteration and model refinement. When considering the performance of models in relation to processing times, it is important to consider the availability of computational resources, the urgency of the results and the relationship between time and accuracy. Some models may offer a good balance between performance and time, while others may require more computational resources and processing time but potentially provide more accurate results. Choosing the ideal model will depend on the user's specific needs and constraints. KNN model with Database 1 may be suitable as faster processing model when time is a critical constraint, as long as model accuracy is acceptable for forecasting needs.

The Adaboost algorithm has its advantages. It has a few hyperparameters that need to be adjusted. This makes its implementation and use simpler compared to other ML algorithms, such as GB, which hyperparameter optimization is a complex task. Besides, it is not prone to overfitting. Nevertheless, Adaboost models with large number of estimators are very time consuming. Training the models required longer execution times and greater computational capacity as the number of estimators increased. If accuracy is the top priority and processing time is less critical, Adaboost and GB xgboost are the best options. Both models performed solidly on both databases. Although the processing time is longer compared to KNN models, the results are worth the time investment. With Database 2, models with longer processing times, such as GB xgboost, GB scikit-learn, and RF, could be used as alternatives to Adaboost model, when accuracy is a higher priority than execution time. Running with Database 1, algorithms like GB xgboost and AdaBoost provide excellent accuracy but may require longer processing times and more computational resources. On the other hand, KNN offers faster processing but with slightly less accuracy. The choice of ML algorithm should consider the trade-off between accuracy and processing time.

VII. Conclusion

7.1. Key Findings

This study focused on using ML with Orange Data Mining to predict CO₂ emissions from a fleet of oil tankers. The models were trained and tested using technical specifications and/or historical voyages data, leading to accurate predictions of CO₂ emissions.

Accurate predictive models were built. Adaboost models presented better performances with both databases. Using technical and voyages database, the best model predicted CO₂ emissions with MAE of 19.66, MAPE of 8.74%, R2 of 0.991, and RRMSE of 19.82%. Using only voyages database, the best model had MAE of 21.78, MAPE of 11.23%, R2 of 0.985, and RRMSE of 24.77%. Comparing with bottom-up approach, Adaboost models had much better performance. In addition to Adaboost models, GB xgboost models also demonstrated accurate predictions for CO₂ emissions with both databases.

The high levels of models' accuracy validate the effectiveness of advanced modeling techniques in estimating CO₂ emissions in the maritime sector, without using weather and sea conditions data. These models present an alternative which diverges from traditional models that rely on *noon reports* and AIS data.

Besides, the use of only ships voyages data, in the form of bulletins sent by the captain at the end of each voyage, provides another alternative approach to CO₂ emissions forecasts that offer reliable performance. By focusing on voyage-related factors and omitting technical specifications, these models can simplify the analysis process and make emission predictions accessible for stakeholders that have access to this kind of data. This approach can be particularly valuable when technical ship data is limited, paid, or not readily available. By relying solely on ships voyages data, such as speed, distance traveled, and voyage time, the models can capture the operational aspects and patterns that directly impact emissions. These factors are directly related to the day-to-day operations of the ships and have a significant influence on emission levels. Therefore, leveraging voyages data allows for analysis of operational factors that contribute to emissions, without relying on technical specifications that may not always be readily available.

This study shows the value of using ML with Orange Data Mining to predict CO₂ emissions from oil tankers. While traditional methods may also allow stakeholders to perform similar analyses, Orange offers several key advantages. Firstly, Orange provides a simplified, user-friendly interface, and intuitive visual interface that enables stakeholders to explore and analyze data without the need for extensive programming skills. Unlike traditional programming languages that require writing code from scratch, Orange provides a simplified and interactive approach to data analysis and modeling. Stakeholders can import data from various sources, pre-process, clean, explore, and visualize data, identify patterns and trends, and create predictive models using a visual interface, all within a single platform. This ease of use allows stakeholders, even those without a strong technical background, to effectively leverage the power of predictive modeling and data analysis. The practical implications of this study's findings extend to multiple stakeholders in the shipping industry. Orange offers a wide range of pre-built algorithms and ML models specifically designed for data analysis in various domains.

In conclusion, the outcomes of this study contribute to the existing theory, fill gaps in the literature, and provide practical implications for stakeholders, by presenting alternative approaches to predicting CO₂ emissions in the maritime industry.

7.2. Critical Reflection

While excluding weather information and vessel position has advantages, it also limits the models' ability to capture the full range of factors that may influence CO₂ emissions. Weather conditions, for instance, can have a significant impact on fuel consumption and vessel performance. Therefore, the decision to exclude these variables should be carefully considered based on the specific research objectives, available data sources, and the trade-off between simplicity and the desired level of accuracy and comprehensiveness in emissions predictions.

BBMs, such as AdaBoost and GB xgboost, are known for their high predictive accuracy but often lack interpretability. These models operate by utilizing complex algorithms and internal calculations that are not easily explainable to non-experts. The lack of transparency in the model's decision-making process can be a significant weakness, as it limits the ability to understand how and why specific predictions are made. BBMs present challenges in interpreting the relationships between input variables and output predictions. The inability to interpret the model's inner workings can make it difficult to gain meaningful insights into the factors driving CO₂ emissions from oil tankers. This limitation can hinder the development of actionable strategies and limit the ability to address specific issues or target interventions effectively.

When using only voyages data, the exclusion of technical specifications may result in a loss of detailed insights into the ship's characteristics and engine efficiency, which can impact emission levels. The trade-off between data accessibility and the precision of emission estimates needs to be considered based on the specific goals and requirements.

Orange Data Mining provides a user-friendly platform for importing, preprocessing, and analyzing data related to CO₂ emissions. However, the simplicity and automation of these processes may limit the depth of

exploration and analysis that can be conducted. Researchers might miss out on more sophisticated data preprocessing techniques or advanced statistical methods that could uncover additional insights or uncover hidden patterns.

7.3. Recommendations for Future Research

For future studies, it is suggested:

- to include other types of ships such as bulk carriers, container ships, general cargo ships, ferries, etc. to provide a comprehensive analysis of emissions across different vessel categories. This would allow for a more holistic understanding of the maritime industry's impact on carbon dioxide emissions.
- to consider alternative fuel types and investigate the potential use of alternative fuels, such as LNG, biofuels, or hydrogen, in oil tankers, comparing their emissions profiles with HFO to assess their environmental benefits and feasibility for reducing carbon dioxide emissions.

As this study focuses on the emissions while the ships are sailing, future research could consider port emissions and activities: expand the study to incorporate emissions generated during stays in port, as these periods often involve significant fuel consumption and emissions.

By addressing these recommendations, future research can provide a more comprehensive understanding of carbon dioxide emissions from oil tankers and contribute to the development of effective strategies for mitigating their environmental impact.

References

- [1] European Union, *Climate action*, 2023. Retrieved from https://climate.ec.europa.eu/eu-action/climate-strategies-targets/progress-made-cutting-emissions_en#:~:text=The%20EU%20has%20put%20in,AgreementEN%E2%80%A2%E2%80%A2%E2%80%A2.
- [2] IMO, *IMO's work to cut GHG emissions from ships*, 2023. Retrieved from <https://www.imo.org/en/MediaCentre/HotTopics/Pages/Cutting-GHG-emissions.aspx>
- [3] Y. Cao, K. Yin, X. Li, and C. Zhai, Forecasting CO₂ emissions from Chinese marine fleets using multivariable trend interaction grey model, *Applied Soft Computing*, 2021.
- [4] M.S. Reis, R. Rendall, B. Palumbo, A. Lepore, and C. Capezza, Predicting ships' CO₂ emissions using feature-oriented methods, *Applied Stochastic Models in Business and Industry*, 2020, 110–123.
- [5] J. Dean, *Big data, data mining, and machine learning: value creation for business leaders and practitioners* (Hoboken, NJ: John Wiley & Sons, 2014).
- [6] T. Mourouzis, *Certificate in big data in shipping* (London: Lloyd's Maritime Academy, 2022).
- [7] I. Zamana, K. Pazoukia, R. Normana, S. Younessic, and S. Colemanb, Challenges and opportunities of big data analytics for upcoming regulations and future transformation of the shipping industry, *Procedia Engineering*, 2017, 537–544.
- [8] D. Yang, L. Wua, S. Wang, H. Jia, and K.X. Lic, How big data enriches maritime research—a critical review of Automatic Identification System (AIS) data applications, *Transport Reviews*, 2019, 755–773.
- [9] L. Aldous, T. Smith, and R. Bucknall, Noon report data uncertainty, *Proc. Low Carbon Shipping Conference*, London, 2013, 1–13.
- [10] IMO, *Third IMO greenhouse gas study 2014*, 2014. Retrieved from <https://www.imo.org/en/ourwork/environment/pages/greenhouse-gas-studies-2014.aspx>
- [11] T. Emmens, C. Amrit, A. Abdi, and M. Ghosh, The promises and perils of Automatic Identification System data, *Expert Systems with Applications*, 2021, 1–15.
- [12] IMO, *AIS transponders*, 2023. Retrieved from <https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx>
- [13] R. Yan, S. Wang, and Y. Du, Development of a two-stage ship fuel consumption prediction and reduction model for a dry bulk ship, *Transportation Research Part E: Logistics and Transportation Review*, 2020, 1–22.
- [14] C. Gonzalez, B. Lund, and E. Hagestuen, Case study: ship performance evaluation by application of big data, *Proc. 3rd Hull Performance & Insight Conference*, Durham, 2018, 12–14.
- [15] Ø.J. Rødseth, L.P. Perera, and B. Mo, Big data in shipping—challenges and opportunities, *Proc. 15th Int. Conf. on Computer Applications and Information Technology in the Maritime Industries (COMPIT 2016)*, Lecce, 2016.
- [16] K.S. Divya, P. Bhargavi, and S. Jyothi, Machine learning algorithms in big data analytics, *International Journal of Computer Science and Engineering*, 6(1), 2018, 63–70.
- [17] K. Lutz, M. Zacharias, S. Klöver, and T. Hensel, *Machine learning in maritime logistics*, 2020. Retrieved from <https://www.cml.fraunhofer.de/en/press/studies/white-paper--machine-learning-in-maritime-logistics-.html>
- [18] B. Pena, L. Huang, and F. Ahlgren, A review on applications of machine learning in shipping, *Ocean Engineering*, 2020.
- [19] L. Huang, B. Pena, Y. Liu, and E. Anderlini, Machine learning in sustainable ship design and operation: a review, *Ocean Engineering*, 2022, 1–18.
- [20] A. Fan, J. Yang, L. Yang, D. Wu, and N. Vladimir, A review of ship fuel consumption models, *Ocean Engineering*, 2022, 1–17.
- [21] D. Sarkar, R. Bali, and T. Sharma, *Practical machine learning with Python: a problem-solver's guide to building real-world intelligent systems* (Berkeley, CA: Apress, 2018).
- [22] R.H. Merien-Paul, H. Enshaei, and S.G. Jayasinghe, In-situ data vs. bottom-up approaches in estimations of marine fuel consumptions and emissions, *Transportation Research Part D: Transport and Environment*, 2018, 619–632.
- [23] IMO, *Fourth greenhouse gas study 2020*, 2020. Retrieved from <https://www.imo.org/en/OurWork/Environment/Pages/Fourth-IMO-Greenhouse-Gas-Study-2020.aspx>
- [24] R. Yan, S. Wang, and H.N. Psaraftis, Data analytics for fuel consumption management in maritime transportation: status and perspectives, *Transportation Research Part E: Logistics and Transportation Review*, 2021.
- [25] J. Demšar and B. Zupan, Hands-on training about overfitting, *PLoS Computational Biology*, 2021, 1–19.
- [26] Orange, *Data mining: fruitful and fun*, 2023. Retrieved from <https://orangedatamining.com/>
- [27] A. Géron, *Hands-on machine learning with scikit-learn and TensorFlow: concepts, tools, and techniques to build intelligent systems* (Sebastopol, CA: O'Reilly Media Inc., 2017).
- [28] C.J. Willmott and K. Matsuura, Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, *Climate Research*, 2005, 79–82.

- [29] C.D. Lewis, *Industrial and business forecasting methods: a practical guide to exponential smoothing and curve fitting* (London: Butterworth-Heinemann, 1982).
- [30] P.D. Jamieson, J.R. Porter, and D.R. Wilson, A test of the computer simulation model ARCWHEAT1 on wheat crops grown in New Zealand, *Field Crops Research*, 27(4), 1991, 337–350.