IPE 6213: Decision Analysis

Final Project Report

Project Title: Vessel Estimated Arrival Time Prediction using a Tree-based Stacking Approach: A Case Study of the Chattogram Port

Submitted by –

Sunny Md. Saber (0422082005)

Muhammad Asifur Rahman (0423082111)

Kya Zaw Thowai

Department of Industrial and Production Engineering,

Bangladesh University of Engineering and Technology, Bangladesh

1. Introduction

A significant proportion of international commerce relies on ports and maritime operations, and it occupies a pivotal position in today's global economy. The majority of containers globally, over 90%, are transported using maritime vessels (Park et al., 2021). The increased efficiency of port operations leads to faster, seamless, and convenient transportation of goods from one port to another. The estimation of Time to Arrival (ETA) is crucial for predicting the arrival of forthcoming boats and scheduling vessel-specific actions for further procedures. The lack of assurance regarding the arrival times of vessels reduces the dependability of the timetable, leading to a rise in delays and a fall in productivity levels for operators involved in inland transport. When a vessel arrives at or departs from a port, it is necessary to designate a specific route. This route comprises several waypoints located between two ports. These waypoints assist us in determining the estimated arrival time of a vessel. In this study, the estimated time of arrival (ETA) between two waypoints is determined, considering that the destination port will be Chittagong Port, which is the largest and most significant port in Bangladesh.

Since 1888, the Chittagong port has played a crucial role in promoting Bangladesh's economic growth as the country's main port, together with its shore-based infrastructure. The port has a total of 15 versatile jetties, along with specialized ones for handling specific commodities such as oil, clinker, and food grain. These jetties are strategically positioned around nine nautical miles from the Bay of Bengal coastline (S. K. Biswas & A. K. M. Solayman Hoque, 2007). According to Lloyd's, it ranked as the 58th busiest container port in the world in 2019. The number of berths on this port is 19. Annual container volume 3.097M TEUs (2020–21). Annual cargo tonnage is 100M (2019–20) (cpa.gov.bd). The coordinate of this port is 22.313°N 91.800°E. During the period, 11,81,74,160 tons of cargo, 32,55,358 TEUS of containers, and 4,231 ships were handled at the port During FY22. Compared to FY 2020-21, cargo handling was 3.9% in fiscal 2021-2022. (https://www.maritimegateway.com/). In 2023, the total number of vessels that arrived at the CTG port was 4,103. The mean daily vessel arrival rate was 11.24. Therefore, effective planning and operational efficiency can establish an ideal and positive atmosphere. To attain these goals, it is necessary to calculate the estimated time of arrival (ETA) with greater precision and carry out operational activities accordingly.

This study employs Automatic Identification System (AIS) data, which encompasses data on vessel position, directions, and speed, to forecast the expected time of arrival (ETA). Determine the distance between our CTG port and other ports using the Vincenty and Haversine formula. The velocity measured at each point along the journey provides us with the mean velocity of each vessel. Subsequently, the data is fed into multiple machine learning algorithms, such as Linear Regression, Random Forest Regression, Support Vector Regression, Artificial Neural Network, K-Nearest Neighbors, XGBoost, and LightGBM, to choose the most optimal approach. This prediction model considers various factors, such as past data on vessel arrivals and departures, as well as the generic characteristics of the vessels. Based on our current understanding, this study is the first to make predictions about ETA using AIS data. The study aims to make four specific contributions and achieve three objectives:

1) To conduct an extensive literature review to identify the best feasible features, output variables, and models for ETA prediction of Chattogram port
2) To develop a novel hybrid regression stacking model, integrating XGBoost and KNN with the LightGBM model, for ETA prediction of incoming vessels
3) To improve vessel arrival prediction at the Chattogram port using AIS data and ML, consequently efficient port management

2. Literature Review

2.1. ETA Prediction and Related Studies

ETA prediction is a very crucial and important part for the port management because they need to complete initial preparation for bathing, unloading and after logistics issues. The accurate ETA prediction leads to minimizing the time and energy of the port and also helps to save a considerable amount of wealth as well. Machine learning and AI lead us in an era that helps us to solve our problems by using the related algorithms. Different types of algorithms can help us to predict the ETA by training the previous data.

A lot of work is conducted to predict ETA through machine learning which are mostly related to our work. Both AIS and LRIT historical maritime traffic data is collected for Port of Trieste, Italy to predict ETA through a data-driven methodology (Alessandrini et al., 2018). In Busan Port to South Korea A novel path-finding algorithm has been used to predict vessel ETA by AIS data that proposes a data-driven methodology (Park et al., 2021). An ETA prediction conducted to the vessels traveling between Japan and Taiwan which considered the future weather condition (uses Bayesian learning to calculate the voyage speed in consideration of future weather) and using path finding algorithm (Ogura et al., 2021). Another data driven methodology that uses vessel trajectory data to consider different maritime situations depending on the location. In this study AIS data are preprocessed using the trajectory mining technique and based on this, a pathfinding algorithm is applied and arrival time is estimated (Kwun & Bae, 2021).

Valero et al., 2021 used an artificial intelligence Internet of Things (AIoT)-based open-source architecture and (the ETA prediction model has been created by training several regression algorithms, such as decision tree, support vector regression, random forest, or K-nearest neighbors (KNN) with AIS (port authority) and maritime oceanographic data (World Weather Online) which is conducted in port of Valencia, Spain. The best result obtained by the reinforcement learning model (Valero et al., 2022) with a Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) of ETA. In Next, Kolley et al., 2023 is one of the related studies that predict ETA with four machine learning algorithms; the results show that also rather simple Machine Learning approaches are able to reach high forecast accuracy. Though the optimization solution does not lead to a robust solution, it helps to ensure less waiting time for vessels.

Wind speed, wind direction, current vessel speed, current direction, water level, significant wave height, and an area indicator are used to predict ETA by a multi-layer artificial neural network (ANN) which used historical AIS data in German sea ports (Jahn & Scheidweiler, 2018). Another study finds the best mathematical model for short term vessel ETA prediction using Support Vector Machines (SVM), Gradient Boosting (GB), K-Nearest Neighbors (KNN) (Flapper, 2020) that used 3 years simulated and real work datasets. Pan et al., 2021 used RNN-LSTM, Genetic Algorithm-BP model by using one month of AIS data in a port of China. The best result obtained by the reinforcement learning model (Valero et al., 2022) with a Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) of 11.31 and 19.56 minutes respectively. A novel context-aware deep learning approach for inland vessel travel time prediction that utilizes deep learning models to effectively capture the correlation between multiple traffic context information and vessel travel time in Wuhan section of the Yangtze River in China (Fan et al., 2023). A novel predictive algorithm that leverages past voyage route patterns and AIS data, Ship Arrival and Departure Declaration system data, and berth plan data that predict ETA by achieving an average Mean Absolute Error (MAE) of approximately 3 h and 14 min. These results surpass the accuracy of existing ETA data (Yoon et al., 2023). AIS data used to predict ETA in Dongliu section of the Yangtze River in China and the results reveal that the proposed model can achieve a prediction time accuracy of 90.5% for downstream vessels and 88.6% for upstream vessels (Lei et al., 2024).

This study utilizes the CTG port to forecast the estimated time of arrival (ETA) for future vessels. Initially, the distance is computed using the Vincenty and Haversine formula, based on the AIS data. Subsequently, various machine learning algorithms, including Linear Regression, Random Forest Regression, Support Vector Regression, Artificial Neural Network, K-Nearest Neighbors, XGBoost, and LightGBM are employed. The algorithms are trained using historical data from CTG port. Additionally, create an innovative hybrid regression stacking model to forecast ETA. In addition, including feature engineering into our study is highly beneficial as it enables us to manipulate raw data and convert it into meaningful features that can be utilized to develop predictive models using machine learning or statistical modeling. Finally, enhancing the accuracy of ETA prediction and demonstrating the correlation between these mathematical models is what sets our work apart.

2.2. Research gaps, and contributions

To the best of our knowledge, no research has been conducted for the ETA prediction of Chattogram port. In addition, no study has used a hybrid regression model using a stacking approach, integrating XGBoost and KNN with the LightGBM model, for ETA prediction.

Therefore, to address the abovementioned research gaps, the following research objectives have been identified, and these are the contributions of the current research -

1. To develop a novel hybrid regression stacking model for ETA prediction of incoming vessels at Chattogram Port
2. To identify features, output, and algorithms for ETA prediction of the port, and to find out the correlation among the features
3. To develop the vessel trajectories, based on AIS data

2.3. Features, Output Variables, and Algorithm Selection

2.3.1. Features Selection

Feature selection attempts to identify the most relevant variables for prediction (Chu et al., 2024). The predictors, and output variables that are used in relevant literature are summarized in Table 1. However, AIS (Automatic Identification System) data - shore and satellite, Shore-based Radar (SBS), Long Range Identification and Tracking (LRIT), SAR Satellite, Optical Satellite, and other maritime sensor systems are used to navigate vessels around the world (Xiao et al., 2019). Among these data sources, AIS data are easily accessible online. As a result, we intend to use AIS data for our current project, as well as relevant AIS data as input variables:

1) Maritime Mobile Service Identity (MMSI)
2) Departure Time
3) Departure Latitude
4) Departure Longitude
5) Arrival Time
6) Destination Latitude
7) Destination Longitude
8) Average Speed
9) Distance
10) Vessel Type
11) Vessel Length
12) Vessel Width
13) Vessel Draft

Some of these features (average speed over ground (SOG), and distance) can not be directly found in AIS data. These features are extracted by feature engineering (mentioned in the methodology section) by using the Haversine Distance Formula.

Table 1: Features and Outcome Variables Used in Relevant Literature

Paper	Features	Outcome Variable
Lei et al., 2024	Length, Width, Type of vessel, Direction of vessel trajectory, Temporal features (Minute, Hour, Day, Week, year) (the year when the voyage begins), Duration of the voyage, Latitude (start and end points), Longitude (start and end points), Distance of the voyage, Water Depth, Number of vessels in the area, Average Area speed	ETA
Kolley et al., 2023	Maritime Mobile Service Identity (MMSI), Current position (latitudinal and longitudinal coordinates), current speed over ground (SOG), course over ground (COG), Heading, Vessel Type, Vessel status, Drift	ETA
Fan et al., 2023	Inputs contain three aspects, namely, vessel sub-trajectory, traffic interaction context information (conflicts), and indirect information from AIS data (month, week, day/night, vessel type, vessel length, width, and power)	ETA
Yoon et al., 2023	vessel name, vessel position, position timestamp, API call timestamp, heading, ETA, Destination, and Speed Over Course (SOG), Representative Paths of vessels from nearby ports	ETA
Valero et al., 2022	Distance, Longitude, Latitude, Average speed, Acceleration, Vessel length, and width	ETA
Valero et al., 2021	Distance, Longitude, Latitude, SOG, COG, Heading, and Draft	ETA
Park et al., 2021	MMSI, Report date, Longitude, Latitude, SOG, COG, shipping trajectories and travel time	ETA
Kwun & Bae, 2021	Details of Departure Points and Arrival Points	ETA
Ogura et al., 2021	MMSI, Voyage speed, Longitude, Latitude, course (Voyage direction), heading (Vessel direction), Timestamp (Passing time as transit site), Significant wave height, Primary wave mean period, Primary wave direction, North-south wind component, East–west wind component	ETA
Pan et al., 2021	MMSI, receiving time, longitude, latitude, speed and heading angle	ETA

Flapper, 2020	Time (Month, Day, Hour), Vessel details (Width, Length, Depth, Type), Previous waypoint (ID, Longitude, Latitude, Distance to current waypoint, Travel time to current waypoint), Current waypoint (ID, Longitude, Latitude, Distance to target waypoint), Target waypoint (ID, Longitude, Latitude)	ETA
Alessandrini et al., 2018	A set of MMSI numbers, vessel tracks, and a geo-referenced polygon of the port area of interest	ETA
Jahn & Scheidweiler, 2018	Wind speed, wind direction, current vessel speed, current direction, water level, significant wave height, and an area indicator	ETA

2.3.2. Output Variable Selection

Since we aim to improve vessel arrival prediction, our output variable will be the "ETA (Estimated Time of Arrival)" of vessels. The relevant literature (Table 1) has also considered ETA as an output variable for their methods.

2.3.3. Algorithm Selection

When developing a berthing schedule for a port, the arrival time for each vessel must be estimated for the planning horizon. This can be accomplished by utilizing current times as a baseline and treating the remaining transit time to the designated port as a regression problem (Jahn & Scheidweiler, 2018). Therefore, the vessels' ETA prediction for a port is a regression problem. Current literature addresses this regression problem using several algorithms (Table 2).

The existing vessel travel time prediction models (Park et al., 2021, Kwun & Bae, 2021, Ogura et al., 2021, Alessandrini et al., 2018) predominantly use path-finding algorithms and corresponding distance/speed relationships to determine journey time (Fan et al., 2023). The primary purpose of this path-finding algorithm is to identify the most efficient route between two geographical sites while minimizing a cost function (Alessandrini et al., 2018). However, these models lack consideration for the complexity of vessel travel time, which is influenced by a variety of traffic-related factors such as collision avoidance, shortest path selection, and vessel personnel performance (Fan et al., 2023).

On the contrary, some of the prediction models (Jahn & Scheidweiler, 2018, Gökkuş et al., 2017) utilized Artificial Neural Network (ANN) for ETA prediction. Nevertheless, Tabular data is the primary data format for port operations and AIS data, with each row demonstrating an observation or sample and each column depicting a feature (Filom et al., 2022). The performance of these tree-based models frequently outperforms that of neural networks in many scenarios, particularly when dealing with tabular data (Lei et al., 2024, Chu et al., 2024). Tree-based models, such as Gradient

Boosting (XGBoost, LightGBM) (Lei et al., 2024, Valero et al., 2022, Flapper, 2020), Random Forests (RF) (Lei et al., 2024, Valero et al., 2022, Valero et al., 2021), and Decision Trees (Kolley et al., 2023, Valero et al., 2022, Valero et al., 2021), are frequently used in predictive analytics. They excel in extracting valuable features and information from datasets using approaches such as bagging and ensemble learning, outperforming neural nets (Grinsztajn et al., 2022).

Additionally, Support Vector Regression (SVR) (Fan et al., 2023, Valero et al., 2021, Flapper, 2020), K-nearest neighbors (KNN) (Kolley et al., 2023, Valero et al., 2022, Valero et al., 2021, Flapper, 2020) have been utilized in many research, and have outperformed many state of the art algorithms. Kolley et al. (2023) also demonstrated that simple machine-learning approaches can achieve high forecasting efficiency in vessel arrival prediction. Therefore, we intend to apply the following machine-learning algorithms to our current project and determine which performs best for the Chattogram port –

1) Linear Regression (LR)

2) Random Forest (RF) Regression

3) Support Vector Regression (SVR)

4) Artificial Neural Network (ANN)

5) K-Nearest Neighbors (KNN)

6) XGBoost

7) LightGBM

8) a hybrid regression (HR) model using a stacking approach, integrating XGBoost and KNN with the LightGBM model

Here, the linear regression model has been employed as a baseline approach. In addition, since XGBoost, LightGBM, and KNN models outperformed other algorithms in terms of individual performance, we integrated them as a hybrid regression model using a stacking approach. The detailed methodologies of these algorithms are outlined in the Methodology section.

Table 2: Algorithms Used for ETA Prediction in Relevant Literature, and Best Algorithm Performance

Paper	Methodology	Best Algorithm Performance
Lei et al., 2024	a hybrid regression model using a stacking approach, integrating XGBoost and RF with the LightGBM model; ANN (BP); RF; XGBoost; LightGBM	A hybrid regression model using a stacking approach, integrating XGBoost and RF with the LightGBM model
Kolley et al., 2023	Linear Regression, KNN, Decision Trees, and ANN	KNN
Fan et al., 2023	A novel context-aware deep learning approach, speed-distance based model (pathfinding algorithm), SVR model, and SPD-LSTM (Speed-Long Short-Time Memory)	A novel context-aware deep learning approach
Yoon et al., 2023	A Novel Predictive Algorithm	A Novel Predictive Algorithm
Valero et al., 2022	Random Forest, Linear Regression, Decision Trees, KNN, and Gradient Boosting	RF
Valero et al., 2021	An AIoT-based open-source architecture (Decision Tree, Support Vector Regression, Random Forest, or KNN)	KNN
Park et al., 2021	A novel path-finding algorithm	A novel path-finding algorithm
Kwun & Bae, 2021	A novel path-finding algorithm, the Dijkstra Algorithm	A novel path-finding algorithm
Ogura et al., 2021	A novel method considering future weather conditions, and another path-finding algorithm	A novel method considering future weather conditions
Pan et al., 2021	RNN-LSTM, Genetic Algorithm-BP model	RNN-LSTM
Flapper, 2020	Support Vector Machines (SVM), Gradient Boosting (GB), and K-Nearest Neighbors (KNN)	Gradient Boosting, SVM
Alessandrini et al., 2018	A novel data-driven path-finding algorithm, Haversine approximation, and land avoidance strategy.	A novel data-driven path-finding algorithm

Jahn & Scheidweiler, 2018	A multi-layer ANN	ANN

3.0. Methodology

The proposed methodology for the research is illustrated in Figure 1. It consists of five essential steps. The initial stage entails conducting an extensive literature review to identify the current practices and research gaps in the prediction scheme for vessels' estimated time of arrival (ETA). Consequently, the features, outcome variables, and potential models for ETA prediction of Chattogram port have been determined. We acquired and preprocessed historical AIS data during the third stage by employing data cleaning, vessel trajectory construction, and feature engineering approaches. Finally, the dataset is divided into training and testing sets, and predictive models are employed to forecast the estimated time of vessel arrival. The final model is developed by hyperparameter tuning, and its performance is assessed by comparing anticipated arrival times to actual values on the test set.

3.1. Data Preprocessing

Preprocessing of the raw AIS dataset is necessary before its utilization in any models, to enhance the quality of the data. The data preparation procedure consists of three essential stages: data cleaning, vessel trajectory development, and feature engineering.

3.1.1. Data Cleaning

Data cleaning is a crucial phase in maintaining the accuracy and reliability of AIS (Automatic Identification System) data, which is mostly comprised of GPS information that is prone to position shifting and missing values. The primary objective of AIS data cleaning is to correct mistakes associated with dynamic position data, such as abnormal track positions, anomalous speeds, and irregular track directions. To address these issues, we are adhering to the following data-cleaning rules outlined in the relevant literature –

- To ensure consistency and uniformity, standardize time formats in AIS databases
- Exclude AIS data points with extremely low speeds or duplicate data entries, which indicate a moored or anchored vessel
- To aggregate all AIS data points for a given vessel, use the Maritime Mobile Service Identity (MMSI) number
- Separate journeys can be identified by looking at time gaps between data points; if the time difference exceeds one hour, one voyage has likely ended and another has begun

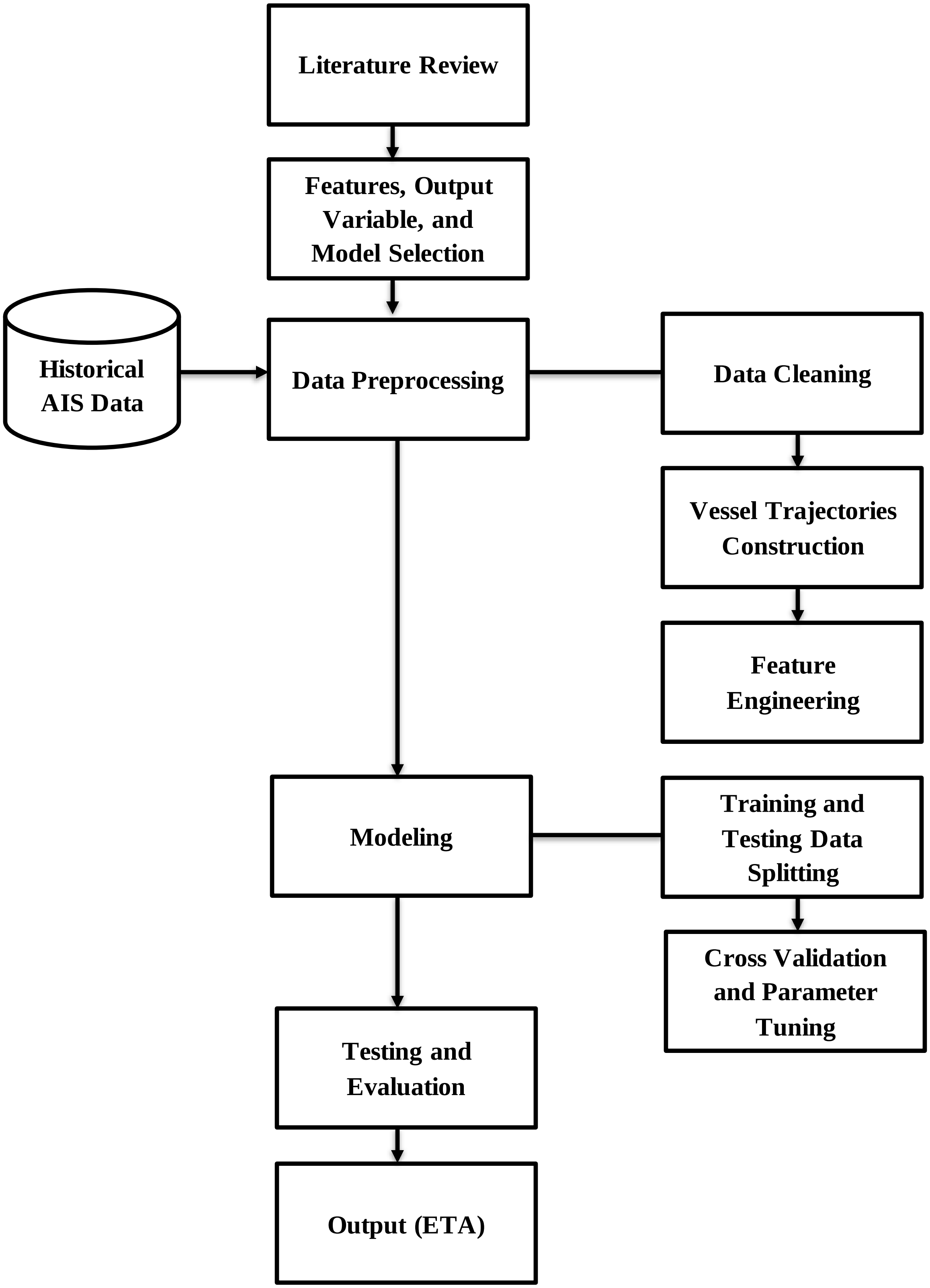
Literature Review
Features, Output Variable, and Model Selection
Historical AIS Data
Data Preprocessing
Data Cleaning
Vessel Trajectories Construction
Feature Engineering
Modeling
Training and Testing Data Splitting
Cross Validation and Parameter Tuning
Testing and Evaluation
Output (ETA)

Figure 1: Methodology Overview

- Eliminate trajectories with durations shorter than 5 minutes or distances shorter than 1 kilometer, as they may be considered anomalies or errors
- Remove trajectories with average velocities that significantly differ from usual values
- Remove AIS messages that have incomplete properties to maintain data completeness and quality

The resulting outcome of the cleaned data is depicted on Figure 2 as vessel trajectories.

3.1.2. Vessel Trajectories Construction

The process of constructing vessel trajectories entails organizing Automatic Identification System (AIS) data into coherent and uninterrupted pathways that represent the movement of vessels over a period of time. The presentation of geographical information in AIS data is commonly in the form of GPS coordinates, which poses difficulties in extracting significant track features or estimating the duration of a voyage. In order to tackle this issue, AIS data points that share the same marine mobile service identification (MMSI) numbers are combined and structured into a GeoDataFrame format. This approach allows for the estimation of parameters such as average speed, distance travelled, and voyage duration for each vessel trajectory. It also makes it easier to choose prior voyages by leveraging the vessel's MMSI and track identification (Lei et al., 2024).

For constructing vessel trajectories, we first converted the raw AIS database to GeoDataFrame format. Then, we utilized ArcGIS software, which offers contextual tools and services for mapping and spatial analysis, to create vessel trajectories, as shown in Figure 2.

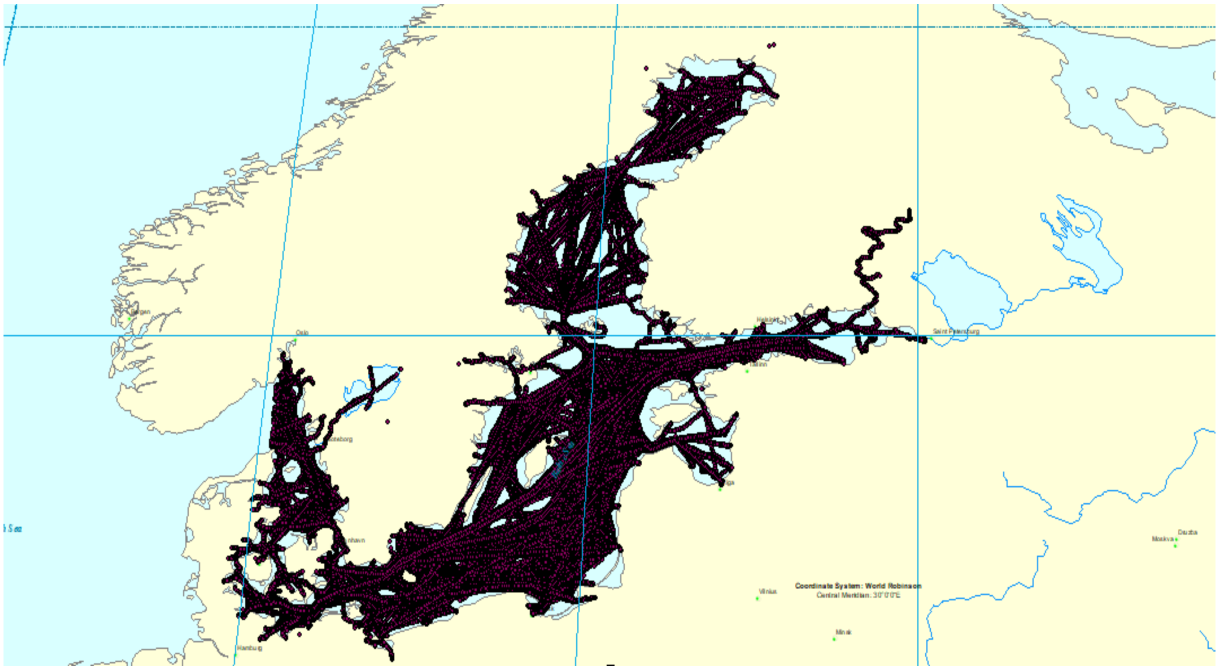

Figure 2: Vessel Trajectories based on AIS data

3.1.3. Feature Engineering

Feature engineering is the process of harnessing domain knowledge to develop more effective representations of an issue in order to improve a model's prediction capabilities (Lei et al., 2024). In this study, vessel-specific information (length, width, and type) was integrated with ship trajectory data to create a comprehensive set of variables for predicting vessel ETAs. However, because speed over ground (SOG) and draft are reflections of weather conditions, and adding weather parameters has no major impact on model performance (Valero et al., 2021), we overlooked meteorological conditions (wind speed, wave direction, etc.) as features. Finally, we determined the value of the distant feature employing the Vincenty Formula. Two sophisticated approaches for estimating coordinate distance are the Haversine Formula and the Vincenty Formula.

The haversine algorithm is a straightforward and effective method for computing the great-circle distance between two sites on the Earth's surface. It assumes that the Earth is a sphere and utilizes the haversine formula to compute the distance between two places on its surface. Before using the haversine formula, the latitude and longitude of each location are converted to radians, as the formula specifically functions with angles in radians. The formula determines the great-circle distance between two places, employing the Earth's radius (in kilometers) to convert the distance from radians to kilometers (Hermawan, 2022; Rooy, 2016b).

Conversely, the Vincenty algorithm provides a more precise technique for computing the great-circle distance between two points on the Earth's surface. It assumes that the Earth is an oblate spheroid (an ellipsoid with slightly flattened poles) and uses a more complicated formula to compute the distance between two places. It first transforms each point's latitude and longitude to radians, as the Vincenty formula requires angles in radians. Subsequently, the formula is employed to compute the great-circle distance between the two places, utilizing the Earth's ellipsoid's semi-minor axis (measured in kilometers) to convert the distance from radians to kilometers (Hermawan, 2022). The Vincenty Formula for distance calculation has been implemented in this research using Python (Rooy, 2016a).

In addition, a correlation matrix has been constructed to analyze the relationship between the features. The results are presented in Table 3. In the correlation matrix, -1 denotes a perfect negative correlation, +1 denotes a perfect positive correlation, and 0 indicates no correlation between the variables.

Table 3: Correlation among the features

	MMSI	DepartureTime	LATd	LONd	ArrivalTime	LATa	LONa	AVGSPDkmph	DistanceKm	VesselType	Length	Width	Draft
MMSI	1												
DepartureTime	0.100	1											
LATd	-0.004	-0.005	1										
LONd	-0.003	-0.025	0.636	1									
ArrivalTime	0.100	0.999	-0.005	-0.022	1								
LATa	0.003	0.100	0.062	0.071	0.103	1							
LONa	0.007	0.084	-0.106	-0.056	0.085	0.716	1						
AVGSPDkmph	0.004	-0.101	-0.135	-0.113	-0.120	0.276	0.391	1					
DistanceKm	0.002	0.094	-0.147	-0.021	0.118	0.355	0.473	0.031	1				
VesselType	-0.003	-0.048	0.091	0.048	-0.038	-0.041	-0.077	-0.067	0.125	1			
Length	0.011	0.002	-0.228	-0.140	0.010	0.357	0.542	0.297	0.463	0.445	1		
Width	0.107	0.004	-0.163	-0.086	0.012	0.395	0.559	0.293	0.483	0.465	0.960	1	
Draft	0.037	0.010	-0.169	-0.090	0.017	0.373	0.532	0.291	0.447	0.513	0.934	0.949	1

3.2. Modeling

The modeling process for predicting travel duration in maritime logistics encompasses a diverse array of regression techniques, each meticulously tailored to exploit the nuances and intricacies of the dataset. The foundational approach, Linear Regression, initiates the analytical journey by predicting travel time based on the input features, employing a straightforward methodology for model training and evaluation. As the complexity escalates, ensemble methods and sophisticated algorithms take center stage, with K-Nearest Neighbor, XGBoost, and LightGBM stepping in to enhance predictive accuracy through their ensemble-based approaches. Complementing these techniques, Support Vector Regressor and Random Forest Regressor contribute their unique methodologies, emphasizing data scalability and proximity-based learning, respectively. Concurrently, Artificial Neural Network introduces a deep learning paradigm, unraveling intricate patterns within the dataset to enable precise predictions. Ultimately, a Stacking Ensemble Model harmonizes the diverse predictions of individual models, culminating in a robust framework capable of tackling the inherent uncertainties in maritime travel forecasting. This comprehensive amalgamation of regression methodologies illustrates a holistic approach towards optimizing prediction accuracy and reliability in maritime logistics, bridging the gap between traditional statistical methods and advanced machine learning techniques. The synergistic interplay of these models not only enhances predictive performance but also offers a multifaceted perspective on the intricate factors influencing travel duration, thereby facilitating informed decision-making and operational efficiency in the maritime domain.

3.2.1. Algorithm Overview

3.2.1.1. Linear Regression

Linear regression is a statistical method used to evaluate the connection between one or more independent variables and a dependent variable. This study utilizes multiple linear regression (MLR) because of its ability to consider several independent variables. By employing Multiple Linear Regression (MLR), a linear function may be represented in the following form:

$$y = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n + \epsilon$$

MLR aims to reduce the error $\in$ of the estimate by fitting the data. This is accomplished through the utilization of the ordinary least squares (OLS) methodology. This strategy utilizes the method of minimizing the squared sum of the errors in order to get the optimal values for β. Put simply, the function that is minimized is the loss function (Vaessen et al., 2021).

$$L(\beta) = \sum_{i=1}^{n} \epsilon_i^2 = \epsilon' \epsilon = (y - X\beta)'(y - X\beta)$$

3.2.1.2. Support Vector Machine or Support Vector Regression

Support Vector Machines (SVM) consider every data point inside a multidimensional space. The dimension of this space is determined by the number of attributes in the data. Organized. Subsequently, the Support Vector Machine (SVM) endeavors to establish a hyperplane in the data space that separates the different classes. If it is not possible to do so, the SVM increases the data space by including more dimensions in a manner that allows for feasibility. The kernel trick is a method of manipulating data as if it were in a higher dimension, without really converting it into that higher dimension. Support Vector Machines (SVMs) may be utilized for both regression and classification tasks. When used in regression, these models are commonly known as Support Vector Regression (SVR) (Flapper, 2020).

3.2.1.3. The Random Forest Regression

Random Forest models may be utilized for both regression and classification problems. In this case, our goal is to predict a continuous variable, namely the time of the cargo train's passage. To do this, we will utilize a Random Forest Regression model. A random forest is an ensemble approach that combines many decision trees, or machine learning algorithms, in order to make predictions. In order to obtain a singular output value, the model calculates the average of the anticipated outcomes for each tree. Figure 3 illustrates the arrangement of a random forest. While this example specifically mentions the presence of 100 trees in the forest, it is important to note that in reality, there can be any number of trees that are considered appropriate.

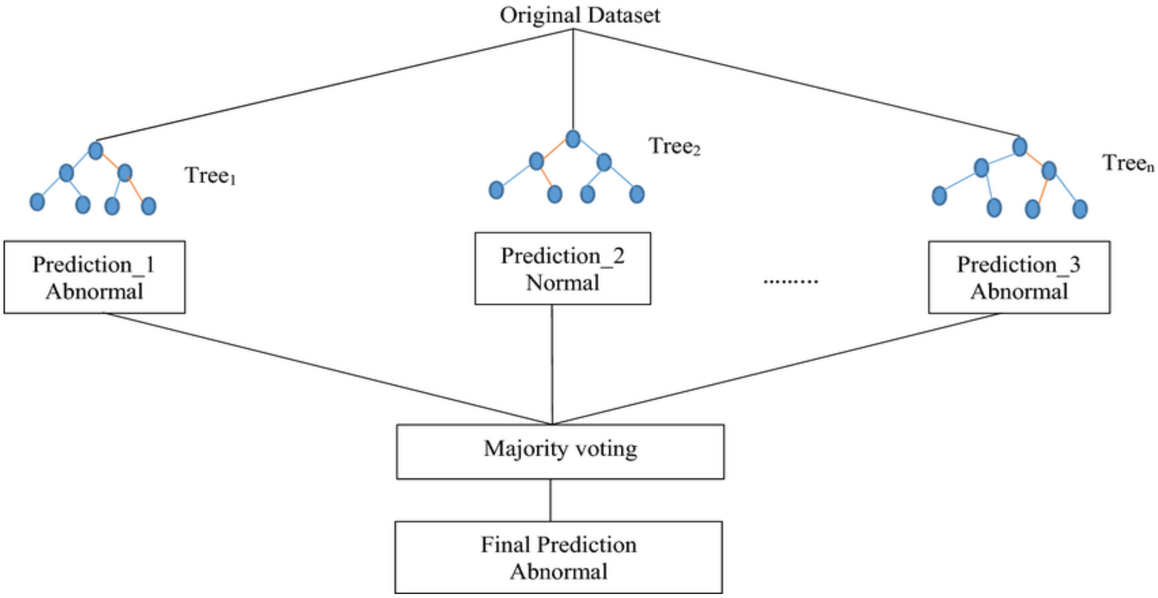

Figure 3: Architecture of a Random Forest

3.2.1.4. Artificial Neural Network

When constructing a functional model of the biological neuron, there are three fundamental components of significance. Initially, the synapses of the neuron are represented as weights. The weight of a connection between an input and a neuron indicates the strength of the connection. In the context of neural networks, negative weight values indicate inhibitory connections, whereas positive values indicate excitatory connections. The following two components simulate the real-time activity occurring within the neuron cell. An adder calculates the total of all the inputs, which have been adjusted by their corresponding weights. The term used to describe this action is linear combination. Ultimately, an activation function governs the magnitude of the output generated by the neuron. The permissible range of output typically falls between the interval of 0 to 1, or alternatively, -1 to 1 (Dongare et al., 2012) Mathematically, this process is described in the Figure 4.

3.2.1.5. K-Nearest Neighbours

K-Nearest Neighbours (KNN) plots each data point into a multi-dimensional space, similar to the Support Vector Machine. KNN however uses these data points when a new set of data arrives. A new data point is placed into the same space and then the nearest points are used to determine the result of the new data. K determines at how many nearest points the algorithm looks. If K is 1 the result will be the result of the nearest point. If K is 10 then the nearest 10 points are used to determine the result (Flapper, 2020).

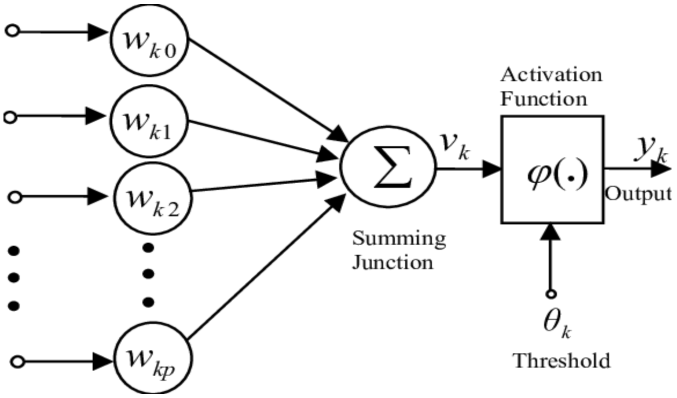

Figure 4: Mathematical Model of ANN

3.2.1.6. XGBoost

XGBoost is an abbreviation for the eXtreme Gradient Boosting program. This package is specifically created and developed to be efficient, adaptable, and portable. The package contains a very effective solution for linear models and an approach for learning trees. The system provides support for a range of objective functions, such as ranking, classification, and regression. The framework is a very efficient and scalable solution that optimizes gradient boosting. A regularized model is employed to manage the complexity of the model, hence simplifying the learning process and preventing overfitting. XGBoost is capable of parallelization. The algorithm employs parallelization during the selection of optimal splitting sites for enumeration, resulting in significantly accelerated training time. The tree construction process is prematurely halted when the prediction results are satisfactory, in order to expedite the training pace. In addition, it allows for the specification of sample weight through the use of the first derivative g and the second derivative h. By modifying the weight, we may provide greater focus to specific samples. Currently, XGBoost stands as one of the most triumphant machine learning methods (ZHAO et al., 2020).

Our ultimate goal is to get an excellent general total model, which minimizes the loss function as much as possible. That is:

$$F_m \arg\ \min \textstyle\sum_{i=1}^{n} L\big(y_i, F_m(x_i)\big) =$$
$$\beta_m \arg\ \min \textstyle\sum_{i=1}^{n} L\big(y_i, F_{m-1}(x_i) + \beta_m f_m(x_i)\big)$$

Solving it concurrently is not feasible due to the ultimate weighting of F by several weak learners. Thus, gradient lifting employs a greedy approach. In the initial stages,

Model F is a constant function that utilizes only one weak learner. The coefficients of this model are determined in a timely manner to gradually enhance the performance of F. Gradient boosting aims to rapidly reduce the loss function by setting the new term equal to the negative gradient of the loss function at each step.

Xgboost improves the regularization learning target L on the basis of the traditional gradient boosting framework. L is:

$$L = \sum_{i} l(\hat{y}_i, y_i) + \sum_{k} \Omega(f_k)$$

3.2.1.7 LightGBM

LightGBM is a distributed boosting system that was introduced by Microsoft DMKT in 2017. Because of its speed and superior performance, it is extensively employed in addressing regression, classification, and other machine learning tasks, particularly in data competitions in recent times.

LightGBM has been demonstrated to be both time-efficient and more precise.

Assuming that a raw dataset with N = {1, 2, …, n} examples and a LightGBM model which has T = {1, 2, …, t} trees are generated. After iterating t times, the final prediction equals the sum of the former (1 − t)th and the th. The iteration process can be described as

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

where y ^(it) is the ith example's prediction value at the tth iteration. f (t) denotes the residuals of the corresponding tree.

3.2.2. Model Formulation

3.2.2.1 Linear Regression

Missing values in the 'ArrivalTime' column were addressed using forward-fill after loading and inspecting the dataset. By subtracting 'DepartureTime' from 'ArrivalTime' timestamps, travel duration in seconds was calculated. An 80-20 split divided the dataset into training and testing sets. The Linear Regression model underwent training on the training data and evaluation on the test set, where metrics like MAE and MSE were computed. Cross-validation assessed the model's generalization ability, yielding cross-validated MAE, MSE, and R-squared scores. Lastly, a scatter plot visualized the comparison between actual and predicted travel durations.

3.2.2.2 Random Forest Regressor

After loading and examining the dataset, missing values were handled through row removal. The data was then split into features (X) and the target variable (y). Subsequent to a train-test split, a RandomForestRegressor model was instantiated. Optimizing parameters such as the number of estimators and maximum depth, hyperparameter tuning occurred via grid search. Cross-validation evaluated model performance, leading to the selection of the best-performing model for predictions. Quantifying the model's performance involved metrics like MAE, MSE, and R-squared.

3.2.2.3 XGBoost

The modeling process commenced with loading the dataset and employing forward-fill to address missing values. Features like departure and arrival times underwent conversion to datetime objects, followed by splitting the dataset into training and testing sets. A grid search approach facilitated hyperparameter tuning, optimizing parameters like maximum depth and learning rate. Evaluating model performance on the test set and through cross-validation utilized metrics such as MAE, MSE, and R-squared. Determining feature importances aided in comprehending the significance of each feature for predicting travel time.

3.2.2.4 LightGBM

Loading the dataset and handling missing values initiated this modeling process. After converting features like departure and arrival times to datetime objects, the dataset underwent splitting for evaluation purposes. Initialization of the LightGBM regressor was followed by grid search-based hyperparameter tuning. Assessment of model performance occurred on both the test set and through cross-validation, employing metrics like MAE, MSE, and R-squared. Scatter plots facilitated understanding the model's predictive accuracy, while feature importance analysis provided insights into significant predictors.

3.2.2.5 Support Vector Regressor (SVR)

Inspecting the AIS data, converting datetime columns appropriately, and removing rows with NaN values to handle missing values marked the start of the SVR modeling process. Relevant features were selected for modeling after converting ArrivalTime into Unix timestamps. Following a training and testing set split, feature scaling took place. Optimal hyperparameters for the SVR model were identified through grid search, after which the model underwent training on the full training data. Evaluation metrics like MAE, MSE, and R-squared assessed the model's predictive performance.

3.2.2.6 K-Nearest Neighbor (KNN)

The KNN modeling process began with loading, inspecting the AIS data, handling missing values, and selecting relevant features for modeling. Splitting the dataset into training and testing sets preceded feature scaling. Hyperparameter tuning via grid search determined the best-performing KNN model, which was then trained and evaluated on the test set. Quantifying the model's predictive performance involved metrics such as MAE, MSE, and R-squared.

3.2.2.7 Artificial Neural Network (ANN)

Loading and inspecting the CSV data initiated the ANN modeling process, followed by handling missing values and converting datetime columns appropriately. After splitting the dataset into training and testing sets, feature scaling took place. Using TensorFlow's Keras library, the neural network model architecture was defined, and the model underwent training on the scaled training data. Metrics like MAE, MSE, and R-squared evaluated the model's predictive accuracy.

3.2.2.8 Stacking Ensemble Model

Preprocessing the AIS data and loading it marked the start of this modeling process. Three regression models (LightGBM, XGBoost, and KNN) were independently trained and evaluated to predict travel time. Combining the predictions of these models formed a stacking ensemble model, whose performance was compared against individual models using various evaluation metrics. Aiding in understanding model performance were visualization techniques like bar plots and scatter plots. Cross-validation ensured the generalization capability of the stacking ensemble model. Ultimately, this approach demonstrated the effectiveness of ensemble techniques in improving predictive accuracy.

3.2.3. Hyperparameter Tuning

Hyperparameter tuning stands as a critical phase in the development of machine learning models, significantly impacting their predictive accuracy and generalization capabilities. This study delves into the profound significance of hyperparameter tuning, elucidating its role in optimizing model performance and enhancing predictive accuracy across various machine learning algorithms. Through meticulous experimentation and systematic exploration of hyperparameter configurations, the study aims to shed light on the intricate interplay between hyperparameters and model performance. The investigation focuses on three prominent machine learning models: K-Nearest Neighbors (KNN), LightGBM, and XGBoost, each representing distinct algorithmic approaches with unique sets of hyperparameters. By employing advanced hyperparameter tuning techniques, such as grid search cross-validation, the hyperparameter space is meticulously navigated to identify optimal configurations that maximize predictive accuracy while mitigating overfitting.

The results of the hyperparameter tuning efforts unveil invaluable insights into the nuanced dynamics between hyperparameters and model performance. For instance, in the case of KNN, the study discerns the optimal number of neighbors and weighting schemes that yield superior predictive accuracy. Similarly, for LightGBM and XGBoost, the research elucidates the criticality of hyperparameters such as learning rate, maximum depth, and the number of estimators in determining predictive efficacy.

3.3. Testing and Evaluation

The testing and evaluation phase is critical for assessing the performance, robustness, and generalization capabilities of machine learning models. A comprehensive framework was implemented to rigorously scrutinize the models and evaluate their real-world applicability. The dataset was partitioned into training and testing subsets, with the latter held out during model training to provide an unbiased evaluation. K-fold cross-validation estimated the generalization error and mitigated overfitting concerns. Evaluation metrics like mean absolute error (MAE), mean squared error (MSE), and R-squared quantified accuracy, precision, and explanatory power on the test set. Visual techniques like scatter plots and residual analysis facilitate identifying prediction patterns, biases, or systematic errors. Feature importance analyses shed light on the significance of input variables, informing future feature engineering efforts. Additionally, the testing process involved assessing computational efficiency, scalability, and robustness to edge cases or anomalous data points. Stress testing and simulations evaluated performance under varying operational conditions, data quality levels, and edge scenarios to ensure reliability and applicability in real-world deployments. The multifaceted approach, combining quantitative metrics, visual diagnostics, feature importance analyses, and robustness assessments, facilitated a thorough understanding of each model's strengths, weaknesses, and practical implications. This comprehensive process enabled informed decision-making regarding model selection, refinement, and deployment strategies within the maritime logistics domain.

4. Experiment

Because there is no open AIS dataset available online for the Chattogram port, we used a dataset of Baltic Sea ports (Hakola, 2020). There are more than 100 ports in the Baltic Sea of Russia, Germany, Poland, Denmark, Finland, Sweden, Lithuania, Latvia, and Estonia (“Ports of the Baltic Sea,” 2023). Figure 5 depicts the map diagram of the Baltic Sea. However, the AIS databases contain 1,048,576 data points from 144 vessels, spanning the period from November 2017 to October 2018. Feature Engineering is utilized to extract 1250 vessel voyage data from the dataset. Afterwards, we divided the dataset into a training set and a validation set with a ratio of 4:1. The training set is utilized to train the model, whereas the validation set is employed to assess the model's performance.

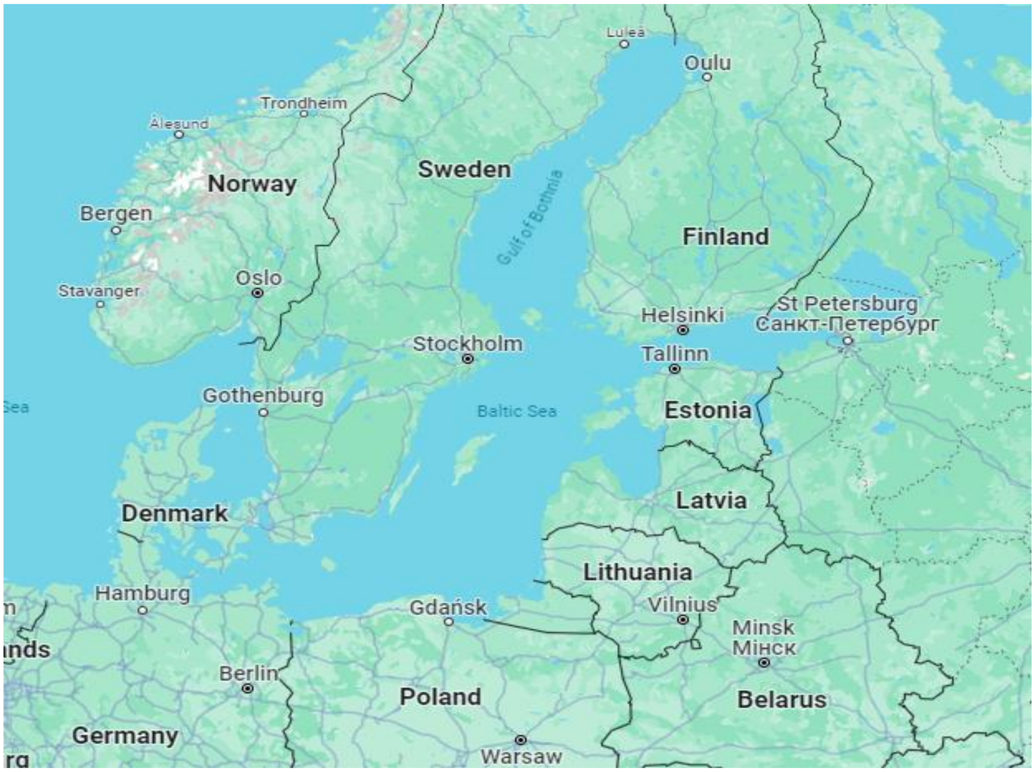

Figure 5: The Map Diagram of the Baltic Sea

5. Results

The Stacking Ensemble Model combining LightGBM, XGBoost, and KNN with Linear Regression as meta-model achieved highest R-squared (0.9949) and lowest mean absolute error (MAE) of 58.9 minutes, outperforming individual models. Cross-validation confirmed its superior generalization (Cross-Val MAE 2 hours 57 minutes, R2 0.9956). XGBoost was next best individual model (R2 0.9958, MAE 55 minutes, Cross-Val MAE 40 minutes, R2 0.9997), followed by LightGBM (R2 0.9933, MAE 4 hours 19 minutes, Cross-Val MAE 3 hours 56 minutes, R2 0.9907). Feature importance analysis revealed LightGBM prioritized average speed, distance, location; XGBoost emphasized distance, average speed, vessel length, draft; both consistently identified average speed and distance as crucial. Visually, Stacking Ensemble had tightest predictions along perfect diagonal, outperforming more dispersed XGBoost, LightGBM, linear regression. Overall, Stacking Ensemble leveraged individual strengths for superior accuracy, followed by XGBoost and LightGBM tree-based models.

5.1.Performance Metrics Comparisons

The comparative analysis of the R-squared (R2) values provides valuable insights into the predictive prowess of the models evaluated in this study. The Stacking Ensemble Model, which integrates the strengths of LightGBM, XGBoost, and K-Nearest Neighbors (KNN) with Linear Regression as the meta-model, demonstrated the highest R2 value of 0.9949080498247753. This exceptional R2 score indicates that the Stacking Ensemble Model can account for an exceptionally large proportion of the variance in the target variable, suggesting its remarkable ability to capture the underlying patterns and relationships within the data.

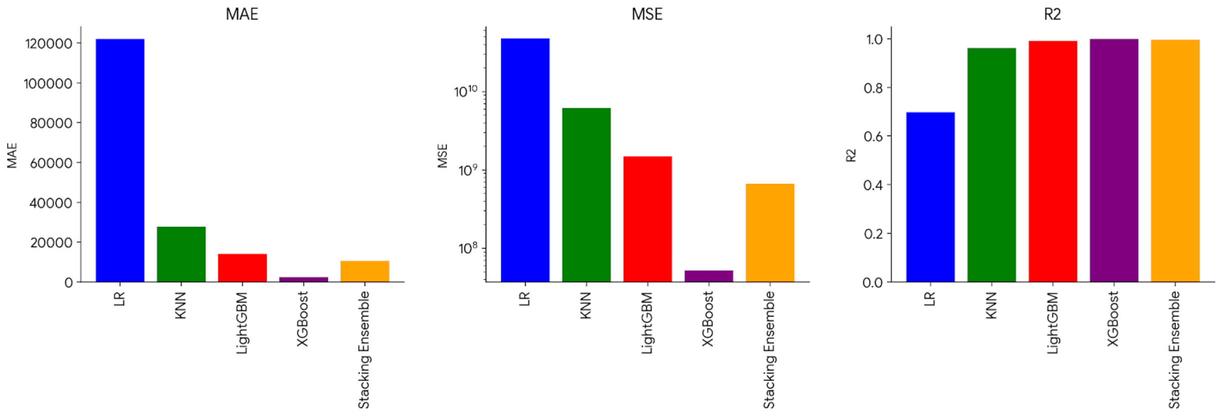

Figure 6: Performance Metrics Comparisons of the Algorithms

In evaluating our Stacking Ensemble Model's performance, we found that it exhibits a mean absolute error (MAE) of 58.9 minutes, reflecting its overall predictive accuracy. Breaking down the contributions of its constituent models, we observe that XGBoost, one of the components, achieves an MAE of 55 minutes, indicating its proficiency in predicting outcomes with a relatively low error margin. Conversely, LightGBM, another individual model within the stack, presents a higher MAE of 4 hours and 19 minutes, signifying a comparatively less precise prediction performance. These insights into the individual model performances provide valuable feedback for fine-tuning our Stacking Ensemble Model and optimizing its predictive capabilities across various scenarios.

Among the individual models, XGBoost emerged as the runner-up, achieving an impressive R2 value of 0.9958575403508116. This finding underscores XGBoost's robust predictive capabilities, as it was able to explain a substantial portion of the target variable's variability. Closely following XGBoost was LightGBM, which delivered a strong R2 score of 0.9933518548318447, further cementing its position as one of the top-performing individual models in this comparative assessment.

In contrast, the remaining individual models, such as Random Forest Regression, Support Vector Regression, and Artificial Neural Network, exhibited significantly lower R2 values, indicating their relative limitations in capturing the nuances and complexities inherent in the target variable. These findings highlight the advantages of the Stacking Ensemble Model and the potential benefits of leveraging ensemble learning techniques to enhance the predictive accuracy and reliability of data-driven models.

Table 4: Performance Metrics Comparisons of the Algorithms

Model	Mean Absolute Error (second)	Mean Squared Error (second**2)	R-squared
Linear Regression	132807.69	4.76e10	0.76
Random Forest Regression	1.53e+18	2.34e+36	-1.95e+22
KNN	34391.87	1.11e10	0.94
LightGBM	15581.64	1.34e9	0.99
XGBoost	3303.60	8.33e8	0.99
Support Vector Regression	9383717.14	1.28e14	-0.07
Stacking Ensemble (LightGBM, XGBoost, KNN)	3534.33	1.02e9	0.99

The superior R2 performance of the Stacking Ensemble Model, even in the rigorous cross-validation process, further corroborates its robust and generalizable predictive capacity. This suggests that the Stacking Ensemble Model is not only effective on the training data but also demonstrates the ability to generalize well to unseen data, making it a highly reliable choice for practical applications across various domains.

5.2 Cross-validation Performance

The cross-validation results provide a comprehensive assessment of the models' generalization capabilities beyond the training data. The Stacking Ensemble Model, which combines LightGBM, XGBoost, and KNN with Linear Regression as the meta-model, exhibited the strongest cross-validation performance. This ensemble-based approach achieved a commendable Cross-Validation Mean Absolute Error (MAE) of 2 hours 57 minutes and an exceptionally high Cross-Validation R-squared (R2) value of 0.9955965259803495, indicating its ability to account for a substantial proportion of the variance in the target variable, even when applied to new, unseen data. Among the individual models, XGBoost delivered impressive cross-validation results, with a Cross-Validation MAE of 40 minutes and a Cross-Validation R2 of 0.9997042892507311, demonstrating its high accuracy and consistency in predictions. LightGBM also performed well in the cross-validation, though not quite at the same level as the Stacking Ensemble Model and XGBoost, with a Cross-Validation MAE of 3 hours 56 minutes and a Cross-Validation R2 of

0.9906997913450845.

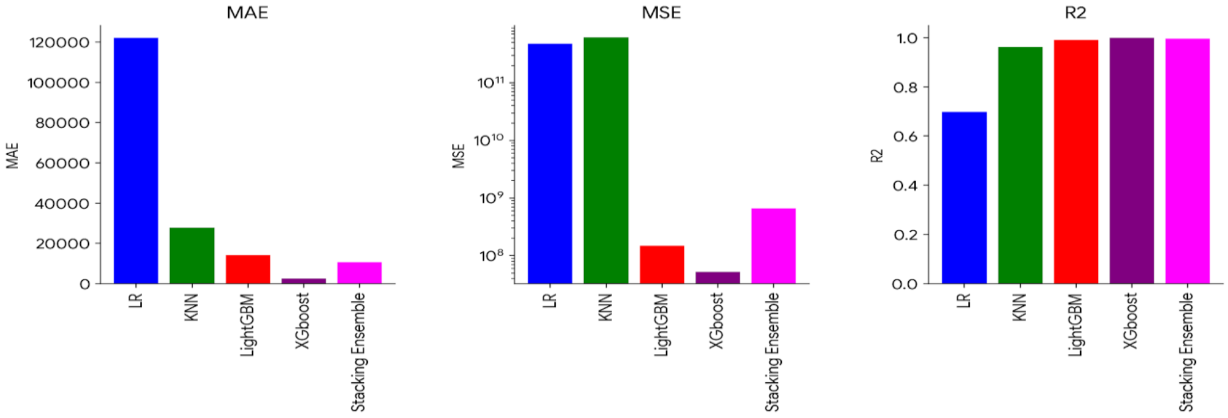

Figure 7: Cross Validation Performance

Table 5: Cross Validation Performance of Models

Model	**Cross-Validation MAE (Second)**	**Cross-Validation MSE (Second**2)**	**Cross-Validation R-squared**
Linear Regression (LR)	122101.19	4.81e10	0.70
KNN	27689.12	6.21e9	0.96
LightGBM	14164.43	1.49e9	0.99
XGBoost	2416.44	5.23e7	0.99
Stacking Ensemble	10644.99	6.63e8	0.99

Overall, the cross-validation findings underscore the superior generalization capabilities of the Stacking Ensemble Model, highlighting its potential for robust and reliable predictions in real-world applications.

5.3 Feature Importance Analysis

The LightGBM feature importance graph (Image 1) reveals that the most important feature is "AVGSPDkmph" (average speed in km/h), followed by "DistanceKm" (distance in km) and "LONa" (a feature related to location). In contrast, the XGBoost feature importance graph (Image 2) shows a different set of feature importances, with "DistanceKm" being the most important, followed by "AVGSPDkmph" and "Length". This disparity in feature importance rankings suggests that the two algorithms have learned different patterns from the underlying data, highlighting the unique dependencies and inductive biases inherent to each model.

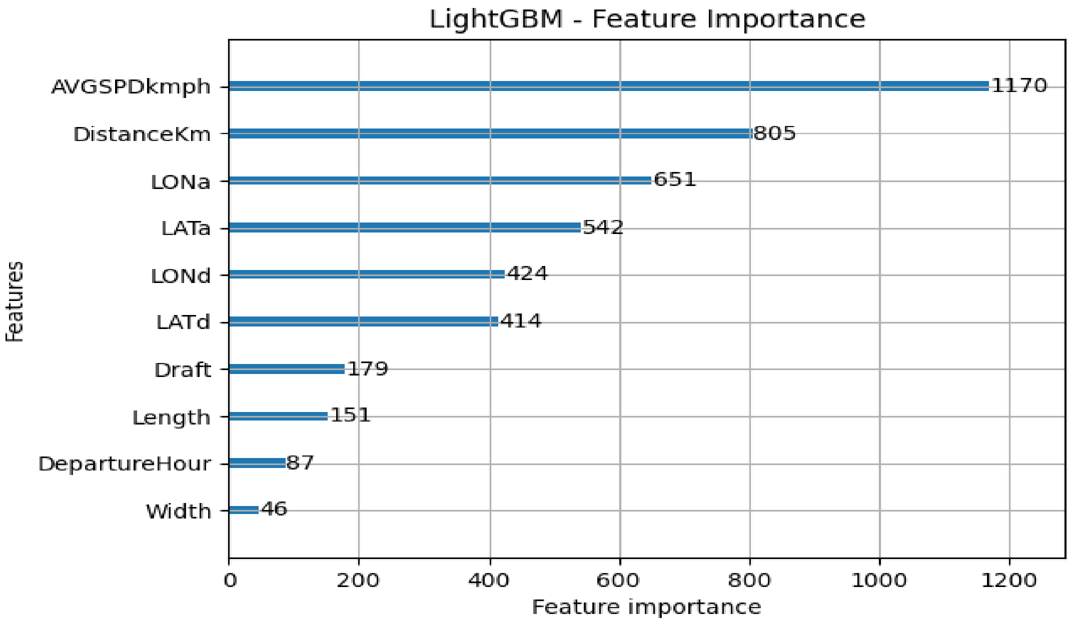

Figure 8: Feature Importance Analysis - LightGBM

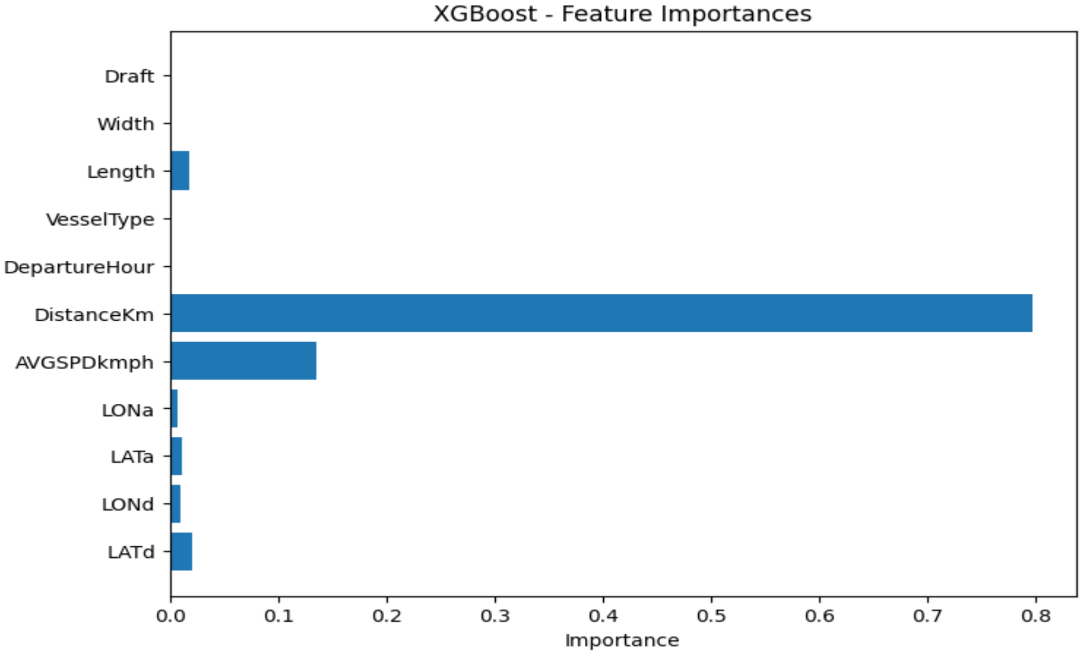

Figure 9: Feature Importance Analysis - XGBoost

5.3.1 Consistent Recognition of Crucial Predictors

Despite the differences in feature importance rankings, both the LightGBM and XGBoost models identify "AVGSPDkmph" and "DistanceKm" as crucial predictors for estimating ETA. This finding underscores the fundamental importance of these high-level factors in determining the arrival time of vessels, irrespective of the specific modeling approach employed. The consistent recognition of these features as highly influential across multiple models reinforces their significance in the domain of ETA prediction and provides a strong foundation for further investigation and model optimization.

5.3.2 Divergent Feature Importance Patterns

While the LightGBM and XGBoost models share a common emphasis on "AVGSPDkmph" and "DistanceKm," the relative importance of other features differs substantially between the two. The LightGBM model considers additional features like "LONa," "LATa," and "LONd" to be important, suggesting that location-based factors play a more prominent role in its predictive capabilities. In contrast, the XGBoost model emphasizes the importance of "Length" and "Draft," indicating that vessel-specific characteristics are more influential in its decision-making process. These divergent feature importance rankings highlight the unique strengths and weaknesses of the two algorithms, and underline the need for a comprehensive, model-agnostic approach to feature importance analysis in order to gain a holistic understanding of the underlying drivers of ETA prediction.

5.3.3 Towards Optimizing ETA Prediction

The present study's comparative analysis of feature importance in LightGBM and XGBoost models for ETA prediction underscores the critical role of this analytical technique in enhancing model interpretability and optimization. While both models identify average speed and distance as key predictors, the specific feature importance rankings differ, reflecting the unique inductive biases and learning mechanisms of the respective algorithms. This diversity in feature importance provides valuable insights that can inform feature engineering, model selection, and ensemble modeling strategies, ultimately leading to more robust and accurate ETA prediction systems. Future research should explore the integration of multiple feature importance analyses, combined with domain-specific knowledge, to further refine and optimize ETA prediction models, thereby advancing the state-of-the-art in transportation and logistics analytics.

5.4 Actual time vs Predicted time (ETA)

In this comparative analysis of different machine learning models for predicting travel times, we observe varying levels of performance across the XGBoost, LightGBM, linear regression, and stacking ensemble approaches. The provided visualizations allow for a thorough examination of the models' predictive capabilities and deviations from the actual travel times. The XGBoost model, an implementation of gradient boosting decision trees, exhibits promising predictive performance. As depicted in the scatter plot, the predicted travel times closely align with the diagonal line representing perfect predictions, indicating a strong correlation between the actual and predicted values. While there are some deviations, the model effectively captures the overall trend, suggesting its suitability for accurate travel time estimations.

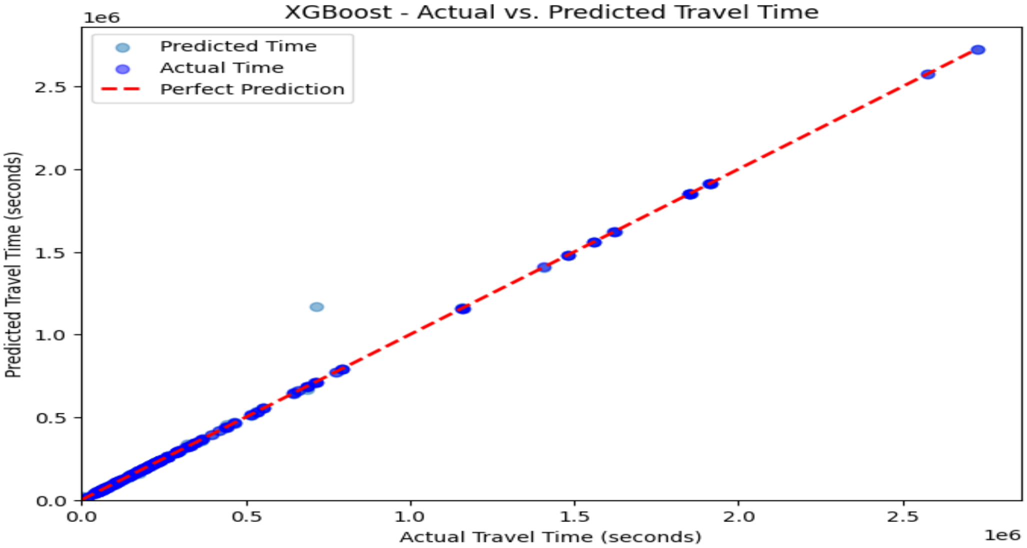

Figure 10: Prediction Accuracy - XGBoost

The LightGBM model, another gradient boosting framework, also demonstrates commendable predictive ability. Similar to XGBoost, the scatter plot illustrates a clear linear relationship between the actual and predicted travel times, with the majority of points clustered around the diagonal line. However, there appears to be a slightly higher degree of dispersion compared to XGBoost, suggesting marginally larger prediction errors in certain instances.

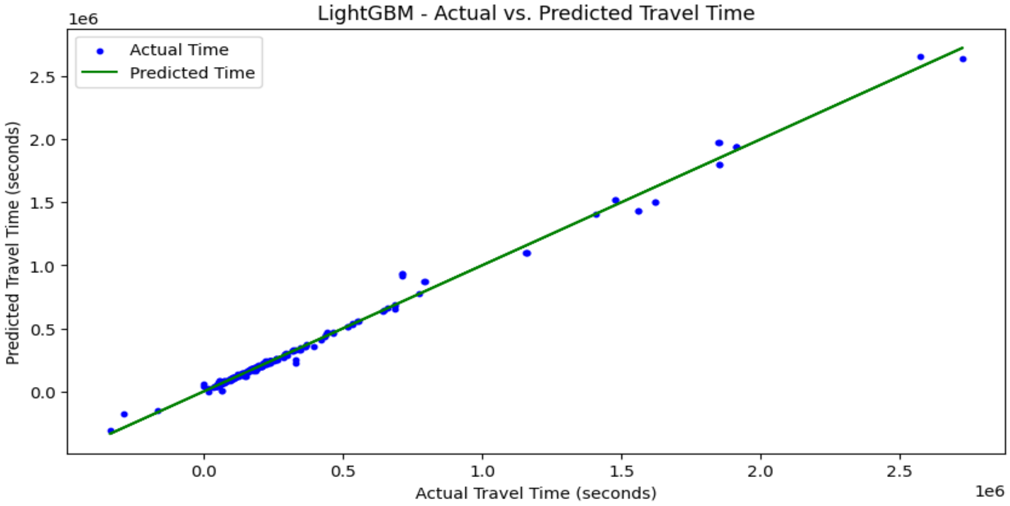

Figure 11: Prediction Accuracy - LightGBM

In contrast, the linear regression model exhibits a more significant deviation from the perfect prediction line, indicating a lower overall predictive accuracy compared to the tree-based models.

The scatter plot reveals a substantial number of points deviating substantially from the diagonal, implying that linear regression may struggle to capture the complex patterns and nonlinearities inherent in the travel time data.

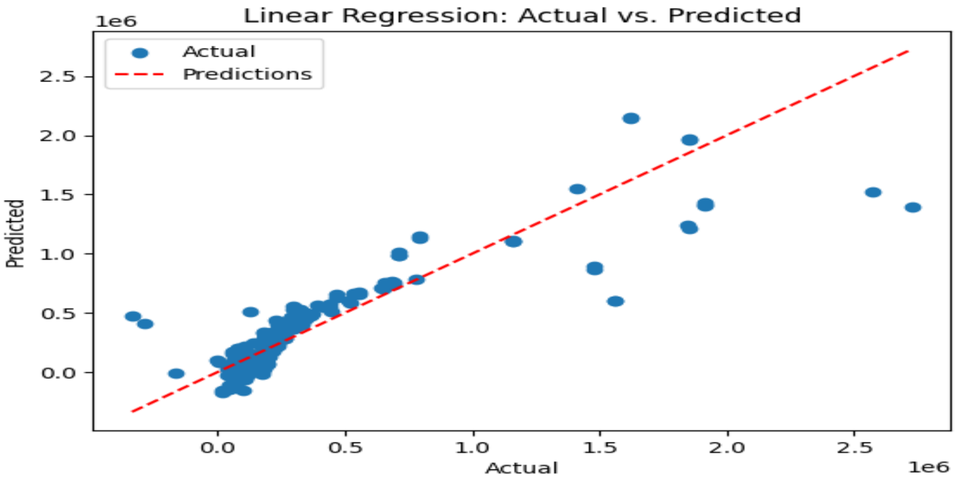

Figure 12: Prediction Accuracy – Linear Regression

The stacking ensemble model, which combines the predictions of XGBoost, LightGBM, and KNN (K-Nearest Neighbors) models, emerges as the most accurate approach among the evaluated techniques. The scatter plot demonstrates an exceptionally close alignment between the predicted and actual travel times, with the majority of points tightly clustered along the diagonal line.

This superior performance can be attributed to the ensemble's ability to leverage the strengths of multiple models, effectively mitigating individual weaknesses and enhancing overall predictive power. Quantitatively, based on the provided information and visualizations, it can be concluded that the stacking ensemble model exhibits the highest accuracy, closely followed by XGBoost, then LightGBM, followed by KNN (not individually visualized), and finally, linear regression, which displays the lowest predictive performance among the evaluated models. It is important to note that the choice of model ultimately depends on the specific requirements of the application, such as the trade-off between accuracy and interpretability, computational efficiency, and the nature of the underlying data. However, in the context of travel time prediction, the stacking ensemble approach and the tree-based models (XGBoost and LightGBM) emerge as robust and highly accurate solutions, outperforming the linear regression model in capturing the complexities of the data.

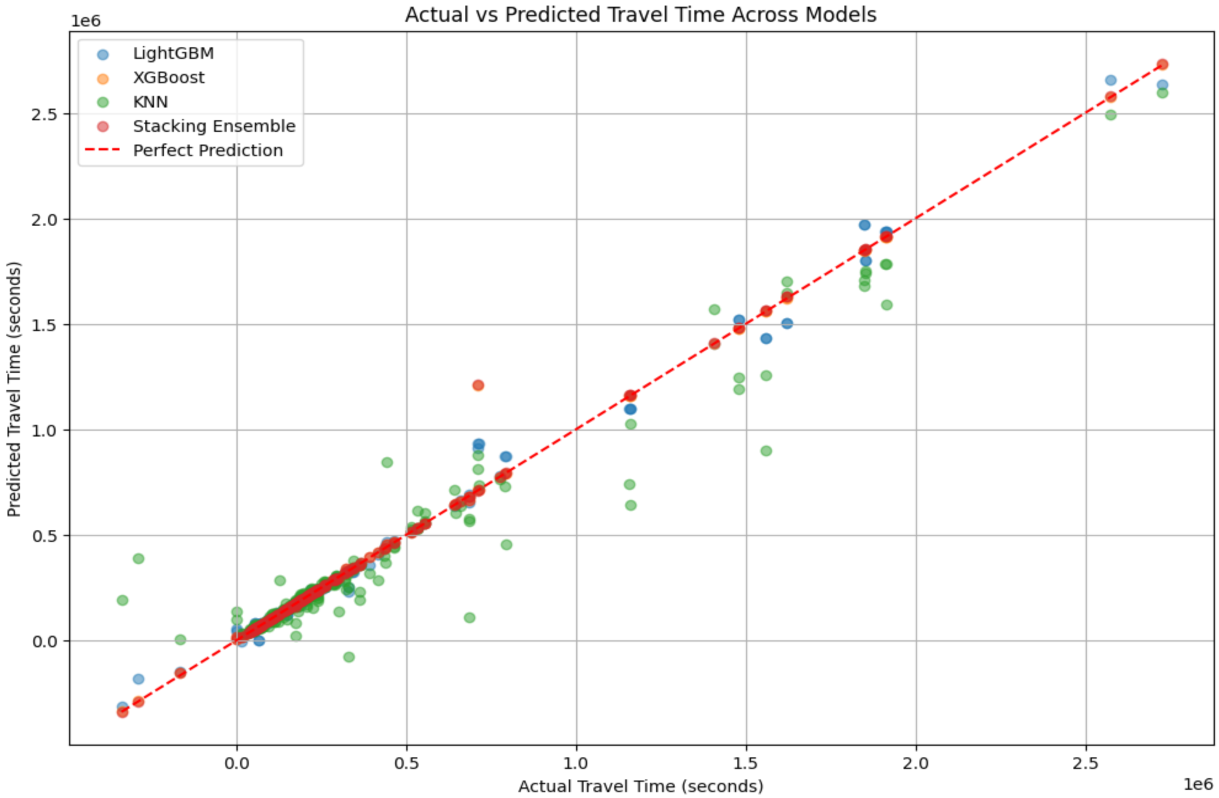

Figure 13: Predictive accuracy of models

6. Discussion

The research endeavor unveils compelling evidence regarding the superior predictive efficacy of the Stacking Ensemble Model, amalgamating LightGBM, XGBoost, and KNN with Linear Regression as the meta-model. This ensemble configuration yielded an impressive R-squared value of 0.9949 and a minimal mean absolute error (MAE) of 58.9 minutes, surpassing the performance of individual models. Particularly noteworthy are its cross-validation outcomes, featuring a Cross-Validation MAE of 2 hours 57 minutes and a Cross-Validation R-squared of 0.9956, showcasing robust generalization capabilities. Moreover, the discerning analysis of feature importance elucidated the consistent significance of average speed and distance as pivotal predictors, with variant rankings highlighting the distinct learning mechanisms intrinsic to each algorithm. Such insights underscore the imperative of meticulous feature analysis in refining ETA prediction systems for enhanced accuracy and interpretability.

6.1 Implications for the Maritime Logistics Industry

Our study underscores the precision advantages of the tree-based stacking model in predicting vessel arrival times using AIS data in the Baltic Sea. However, several limitations necessitate further exploration. Firstly, our research focuses exclusively on the Baltic Sea, limiting the broader applicability of our findings. Moreover, the AIS data utilized may not encompass all relevant factors influencing ETA prediction, such as weather conditions, traffic patterns, wind speed, current speed, and tidal directions. Future investigations should encompass a more comprehensive dataset to account for these factors effectively. Additionally, our reliance on foundational models such as LightGBM, XGBoost, and KNN without introducing novel adaptations may contribute to any observed limitations. Enhancing these foundational models based on the characteristics of AIS data could significantly bolster prediction accuracy.

6.2 Further Research and Research Opportunities

As we delve deeper into the realm of maritime data prediction, the exploration of additional modeling techniques beyond the scope of this study presents a compelling avenue for future research. Incorporating advanced algorithms such as CatBoost, a gradient boosting algorithm known for its robustness to categorical variables, could enhance the predictive capabilities of ETA estimation models in the Baltic Sea region. Furthermore, the application of time series analysis methodologies offers a promising approach to capturing temporal dependencies and trends inherent in vessel trajectory data, thereby improving the accuracy of ETA predictions over time. Moreover, the integration of ensemble learning techniques beyond the traditional stacking approach, such as bagging and boosting, warrants investigation to leverage the collective intelligence of diverse models and enhance prediction robustness. Additionally, the inclusion of external data sources such as satellite imagery, maritime traffic reports, and historical weather data could provide valuable contextual information to augment prediction accuracy and resilience to unforeseen events.

In parallel, exploring the impact of dynamic factors such as changing environmental conditions, evolving traffic patterns, and geopolitical influences on vessel ETA predictions presents a rich area for future inquiry. By incorporating these dynamic elements into predictive models and optimization frameworks, researchers can develop more adaptive and responsive decision support systems tailored to the unique challenges of maritime logistics in the Baltic Sea region. Furthermore, the development of user-friendly visualization tools capable of integrating real-time AIS data, predictive analytics, and route optimization algorithms offers a promising avenue for enhancing situational awareness and decision-making capabilities for vessel operators and maritime authorities. Such tools can empower stakeholders with actionable insights, facilitating proactive risk management and operational planning in the dynamic maritime environment of the Baltic Sea. By exploring a diverse array of modeling techniques, incorporating dynamic external factors, and leveraging advanced visualization technologies, future research endeavors have the potential to revolutionize maritime data prediction and decision support systems in the Baltic Sea region and beyond. These research opportunities underscore the ongoing quest for innovation and excellence in maritime logistics, driving towards a safer, more efficient, and sustainable maritime transportation ecosystem.

Reference

Alessandrini, A., Mazzarella, F., & Vespe, M. (2018). Estimated time of arrival using historical vessel tracking data. IEEE Transactions on Intelligent Transportation Systems, 20(1), 7-15.

Chu, Z., Yan, R., & Wang, S. (2024). Vessel turnaround time prediction: A machine learning approach. Ocean & Coastal Management, 249, 107021.

Dongare, A. D., Kharde, R. R., & Kachare, A. D. (2012). Introduction to artificial neural network. International Journal of Engineering and Innovative Technology (IJEIT), 2(1), 189-194.

Fan, T., Chen, D., Huang, C., Tian, C., & Yan, X. (2023). Inland Vessel Travel Time Prediction via a Context-Aware Deep Learning Model. Journal of Marine Science and Engineering, 11(6), 1146.

Filom, S., Amiri, A. M., & Razavi, S. (2022). Applications of machine learning methods in port operations–A systematic literature review. Transportation Research Part E: Logistics and Transportation Review, 161, 102722.

Flapper, E. (2020). ETA Prediction for vessels using Machine Learning (Bachelor's thesis, University of Twente).

Hermawan, H. (2022, December 7). Comparing the Haversine and Vincenty Algorithms for Calculating Great-Circle Distance. The Medium. https://medium.com/@herihermawan/comparing-the-haversine-and-vincenty-algorithms-for-calculating-great-circle-distance-5a2165857666

Hoque, A. S., & Biswas, S. K. (2007). Berthing problem of ships in Chittagong Port and proposal for its solution. Journal of Mechanical Engineering, 37, 66-70.

Jahn, C., & Scheidweiler, T. (2018). Port call optimization by estimating ships’ time of arrival. In Dynamics in Logistics: Proceedings of the 6th International Conference LDIC 2018, Bremen, Germany (pp. 172-177). Springer International Publishing.

Kolley, L., Rückert, N., Kastner, M., Jahn, C., & Fischer, K. (2023). Robust berth scheduling using machine learning for vessel arrival time prediction. Flexible Services and Manufacturing Journal, 35(1), 29-69.

Kwun, H., & Bae, H. (2021). Prediction of vessel arrival time using auto identification system data. Int J Innov Comput Inf Control, 17(2), 725-734.

Lei, J., Chu, Z., Wu, Y., Liu, X., Luo, M., He, W., & Liu, C. (2024). Predicting vessel arrival times on inland waterways: A tree-based stacking approach. Ocean Engineering, 294, 116838.

Ogura, T., Inoue, T., & Uchihira, N. (2021). Prediction of arrival time of vessels considering future weather conditions. Applied Sciences, 11(10), 4410.

Pan, N., Ding, Y., Fu, J., Wang, J., & Zheng, H. (2021, June). Research on Ship Arrival Law Based on Route Matching and Deep Learning. In Journal of Physics: Conference Series (Vol. 1952, No. 2, p. 022023). IOP Publishing.

Ports of the Baltic Sea. (2023, August). In Wikipedia. https://en.wikipedia.org/wiki/Ports_of_the_Baltic_Sea

Rooy, N. (2016a, December 18). Calculate the Distance Between Two GPS Points with Python (Vincenty's Inverse Formula). GitHub. https://nathanrooy.github.io/posts/2016-12-18/vincenty-formula-with-python/

Rooy, N. (2016b, September 7). Calculating the Distance Between Two GPS Coordinates with Python (Haversine Formula). GitHub. https://nathanrooy.github.io/posts/2016-09-07/haversine-with-python/

Vaessen, M. G., Dugundji, E. J., & Bhulai, S. (2021). Predicting the ETA of cargo trains using AI models.

Valero, C. I., Ivancos Pla, E., Vaño, R., Garro, E., Boronat, F., & Palau, C. E. (2021). Design and development of an aiot architecture for introducing a vessel eta cognitive service in a legacy port management solution. Sensors, 21(23), 8133.

Valero, C. I., Martínez, Á., Oltra-Badenes, R., Gil, H., Boronat, F., & Palau, C. E. (2022). Prediction of the Estimated Time of Arrival of container ships on short-sea shipping: A pragmatical analysis. IEEE Latin America Transactions, 20(11), 2354-2362.

Ville Hakola. (2020). Vessel tracking (AIS), vessel metadata and dirway datasets. IEEE Dataport. https://dx.doi.org/10.21227/j3b5-es69

Xiao, Z., Fu, X., Zhang, L., & Goh, R. S. M. (2019). Traffic pattern mining and forecasting technologies in maritime traffic service networks: A comprehensive survey. IEEE Transactions on Intelligent Transportation Systems, 21(5), 1796-1825.

Yoon, J. H., Kim, D. H., Yun, S. W., Kim, H. J., & Kim, S. (2023). Enhancing Container Vessel Arrival Time Prediction through Past Voyage Route Modeling: A Case Study of Busan New Port. Journal of Marine Science and Engineering, 11(6), 1234.

Zhao, W. P., Li, J., Zhao, J., Zhao, D., Lu, J., & Wang, X. (2020). XGB model: Research on evaporation duct height prediction based on XGBoost algorithm. Radioengineering, 29(1), 81-93.