Energy Consumption Prediction for Ro-Ro Vessels Using Machine Learning Approaches

Mete Harun Akcay

I. Introduction

Energy management is an important concept, especially in maritime operations where access to energy is limited. Ensuring that a ship has enough stored energy and accurately tracking energy consumption is crucial for maintaining efficiency and preventing unexpected shortages. Predicting energy consumption allows better planning, fuel optimization and cost reduction. This study focuses on utilizing machine learning models to predict the energy consumption of a ship based on sensor data collected during its voyages.

The data for this study was obtained from the Danish Ro-Ro passenger ship MS Smyril. This vessel has a length of 135 m, a width of 22.7 m, and a design draft of 5.6 m. It is powered by four 3,360 kW main engines. The ship's operational data was collected from February to April 2010, covering 246 voyages and recording a total of 1,627,324 data entries. The dataset includes measurements from multiple sensors, such as the Doppler speed log, gyrocompass, Global Positioning System (GPS), main energy pipe flow meter, rudder angles, wind, propeller pitches, inclinometer, and level measurement device.

The goal of this study is to use these sensor data to make an hourly-based energy consumption prediction. The remainder of this paper is structured as follows: Section II describes data processing, including feature selection, data cleaning, and feature engineering. Section III presents an exploratory data analysis of the features and their relationships with energy consumption. Section IV covers the machine learning models used, their training and evaluation process, and hyperparameter tuning. Finally, Section V discusses key findings, challenges encountered, and potential improvements for future work.

II. Data Processing

The dataset consists of 20 CSV files, each containing two columns: ID (a numerical timestamp in a unique format) and a sensor measurement. Since merging all files at once led to excessive RAM usage in Google Colab, an incremental processing approach was adopted. Each dataframe was processed individually before merging them into a final dataset.

For each dataframe, the ID column was converted into a human-readable datetime format to allow proper alignment of sensor readings. To manage the large volume of data, each dataframe was aggregated into hourly intervals, reducing the number of rows from millions to a range of 680 to 710 in various dataframes. Different aggregation techniques were applied based on the nature of the data:

- Most numerical sensor readings were aggregated by their mean value per hour.
- Sensor measurements recorded in degrees (e.g., true heading, wind angle) were aggregated using circular mean.
- Distance-related parameters were derived from speed measurements. Since timestamp intervals were not uniform, a new column, `time_diff`, was introduced to represent the time difference between consecutive readings. Each row's distance traveled was then computed using:

$$\text{distance} = \text{speed} \times \text{time\_diff}$$

- To track total ship movement, latitude and longitude differences (`lat_diff` and `long_diff`) were computed, capturing total displacement over each hourly period.

Then, the target column, Energy Consumption (EC) was calculated using fuel-related parameters. The hourly fuel consumption rate was calculated using:

$$EC = \frac{(\text{fuelDensity} \times \text{fuelVolumeFlowRate}) \times 3600}{1000} \quad \text{(tons/hour)}$$

Then, all the dataframes were merged based on their Hour values. After merging, all rows containing NaN values were removed, resulting in a final dataframe with 683 rows and 26 columns. The dataframe spans from 2010-02-16 12:00:00 to 2010-04-12 22:00:00.

III. Data Analysis

After merging the dataframes, two additional columns were introduced to explore potential patterns in energy consumption: hour (ranging from 0 to 23) and day (ranging from 0 to 6, representing the day of the week). These features were added to investigate whether energy consumption exhibits any trends based on the time of day or the day of the week.

Figures 1, 2, and 3 present key insights into energy consumption patterns.

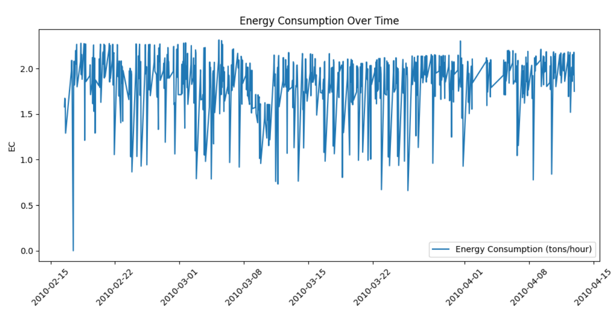

Fig. 1: Energy consumption over time.

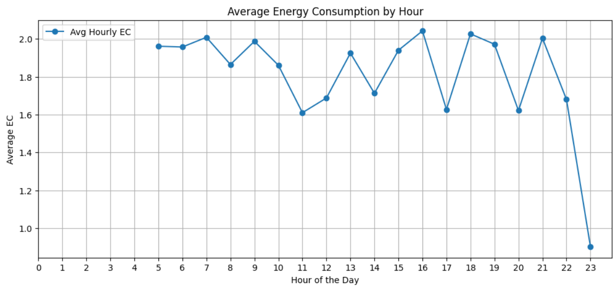

Fig. 2: Average energy consumption by hour of the day.

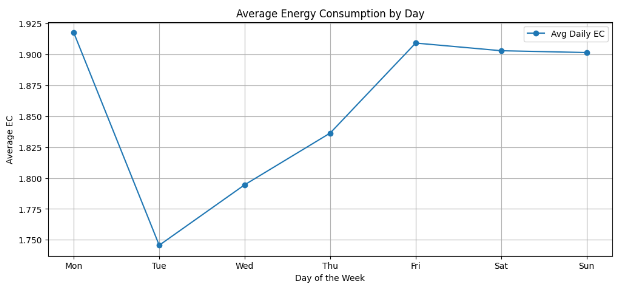

Fig. 3: Average hourly energy consumption by day of the week.

From the graphs, it can be observed that energy consumption generally fluctuates between 2.3 and 1.6 tons per hour, but occasionally drops below 1. When analyzing energy consumption by hour, it appears that this drop occurs around 23:00, while consumption remains relatively stable throughout other hours of the day. When examining daily trends, energy consumption exhibits an increasing pattern from Tuesday to Friday. On Saturday, Sunday, and Monday, the values remain as high as those observed on Friday.

Since the dataset contained a large number of features, an initial feature selection process was applied to improve model interpretability and performance. First, features with a correlation lower than 0.3 with EC were removed. Additionally, fuelVolumeFlowRate and fuelDensity were excluded, as they were directly used in the calculation of EC.

After filtering, the remaining features were sorted in descending order based on their correlation with EC, as shown below:

Feature	Correlation with EC
portPitch	0.925090
starboardPitch	0.903612
longitudinalWaterSpeed	0.903090
speedKmh	0.897678
speedKnots	0.897677
portRudder	0.761536
starboardRudder	- 0.719028
longitudeMovement	0.669223
latitudeMovement	0.663742
distanceWater	0.660123
distance	0.647087
level2median	0.614292
Longitude	0.488213
windSpeed	0.358079

TABLE I: Features sorted by correlation with EC.

However, some features exhibited very high correlation with each other, which could lead to redundancy and multicollinearity in the model, ultimately resulting in distortion of model coefficients and a decrease in generalization performance. To avoid this issue, highly correlated features were removed based on the following rationale:

- distance and distance_water were removed due to their 97% correlation with latitude_movement, and since latitude_movement had a higher correlation with EC, it was retained.
- speedKmh, speedKnots, and longitudinalWaterSpeed were removed because they were all 97% correlated with portPitch, and portPitch was more strongly correlated with EC.
- Although portPitch and starboardPitch were also 97% correlated with each other, they were retained due to the lack of domain knowledge regarding whether one is more informative than the other.

After removing redundant and highly correlated features, 10 features remained for modeling. The heatmap in Figure 4 illustrates the correlation between these remaining features.

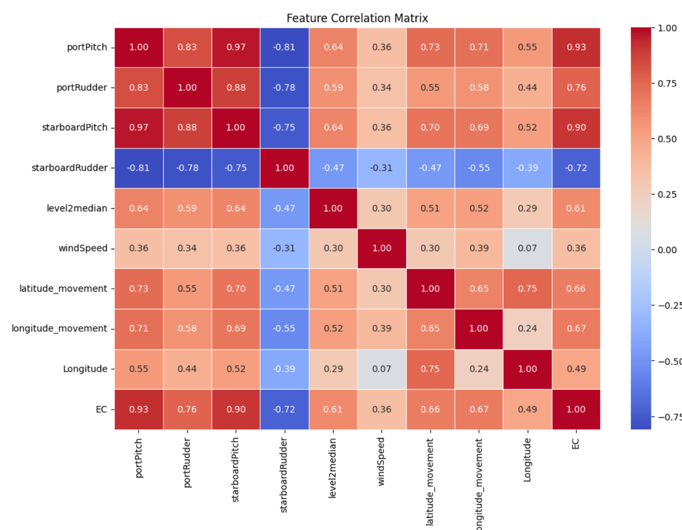

Fig. 4: Heatmap showing the correlation between the remaining 10 features.

The scatter plots in Figure 5 illustrate the relationships between EC and the selected features. The key observations are as follows:

- There is a strong positive correlation between pitch values and EC, indicating that as the pitch value increases, EC also increases. This aligns with expectations since a higher pitch generally means more thrust, leading to greater fuel consumption.
- While portRudder has a high positive correlation, a high negative correlation is observed with starboardRudder, suggesting that higher rudder angles tend to lower EC.
- A moderate positive correlation is present between level2median and EC.

- There is a mild positive correlation with windSpeed and EC. A new feature named windForce, combining windSpeed and windAngle, was also calculated; however, windSpeed was more correlated to EC.
- There is a strong positive correlation between latitude, longitude movements and EC. This is expected, as larger positional changes imply more distance traveled, naturally requiring more fuel.
- A weak trend is observed where certain longitude values correspond to lower EC. This may be due to specific operational conditions or routes where fuel efficiency is higher.

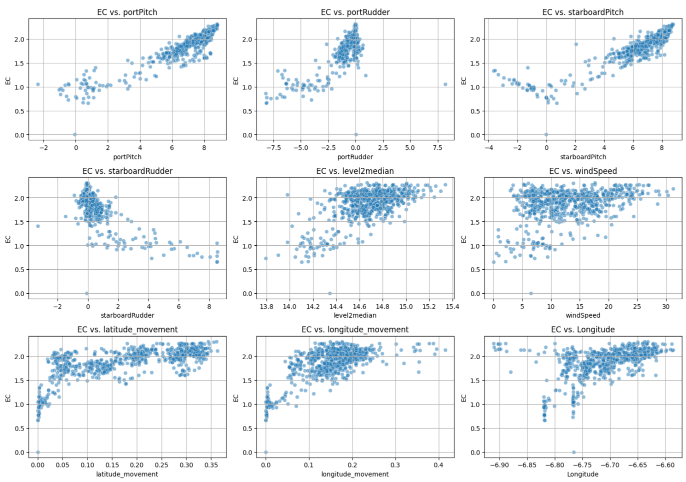

Fig. 5: Scatter plots showing the relationship between EC and selected features.

IV. Modeling

After finalizing the selection of features, the dataset was split into 80% training, 10% validation, and 10% testing to ensure that the models could generalize well to unseen data. Before training, feature scaling was applied to normalize the input variables. Scaling was necessary because some features had different numerical ranges, which could affect the performance of certain models, especially those relying on distance-based calculations (e.g., Support Vector Machines). Standardization was used to transform all features to have a mean of zero and a standard deviation of one.

Six different machine learning models were trained using their default parameters to establish baseline performance. The evaluation metrics, R² score and Root Mean Squared Error (RMSE), are presented in Figures 6 and 7.

Though all the trained models exceeded the target accuracy 85%, Random Forest Regressor and Gradient Boosting Regressor achieved the highest performance with the following scores:

- Gradient Boosting: $R^2 = 0.9449$, RMSE = 0.0078
- Random Forest: $R^2 = 0.9411$, RMSE = 0.0083

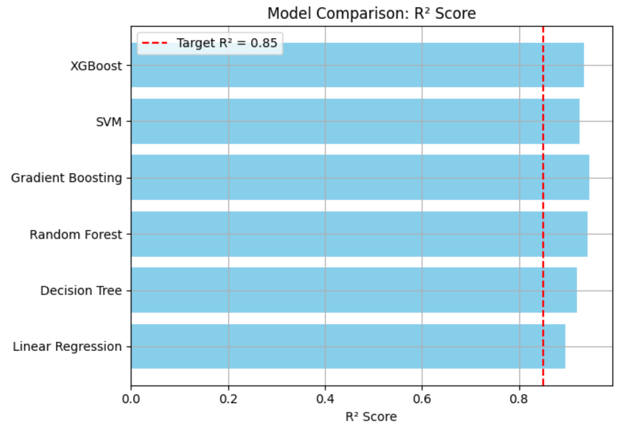

Fig. 6: R² scores of different models.

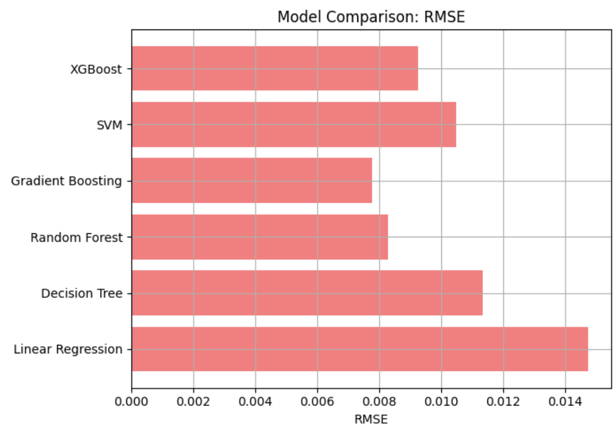

Fig. 7: RMSE values of different models.

Given their superior performance, these two models were selected for further hyperparameter tuning to optimize their predictive accuracy. For this, Grid Search was applied on the hyperparameters of Gradient Boosting and Random Forest models. The following hyperparameter grids were defined:

For the Gradient Boosting model:

- n_estimators: [50, 100, 200] (Number of boosting stages)
- learning_rate: [0.01, 0.1, 0.2] (Step size shrinkage to prevent overfitting)
- max_depth: [3, 5, 10] (Maximum depth of individual trees)
- min_samples_split: [2, 5, 10] (Minimum number of samples required to split an internal node)
- min_samples_leaf: [1, 2, 4] (Minimum number of samples required to be at a leaf node)

For the Random Forest model:

- n_estimators: [50, 100, 200] (Number of decision trees in the forest)
- max_depth: [None, 10, 20]
- min_samples_split: [2, 5, 10]
- min_samples_leaf: [1, 2, 4]

After performing grid search with cross-validation, the best hyperparameters were found to be:

- Best Gradient Boosting Parameters: {learning_rate: 0.1, max_depth: 5, min_samples_leaf: 2, min_samples_split: 2, n_estimators: 200}
- Best Random Forest Parameters: {max_depth: None, min_samples_leaf: 2, min_samples_split: 2, n_estimators: 200}

Figure 8 presents the validation and test performance of the optimized models.

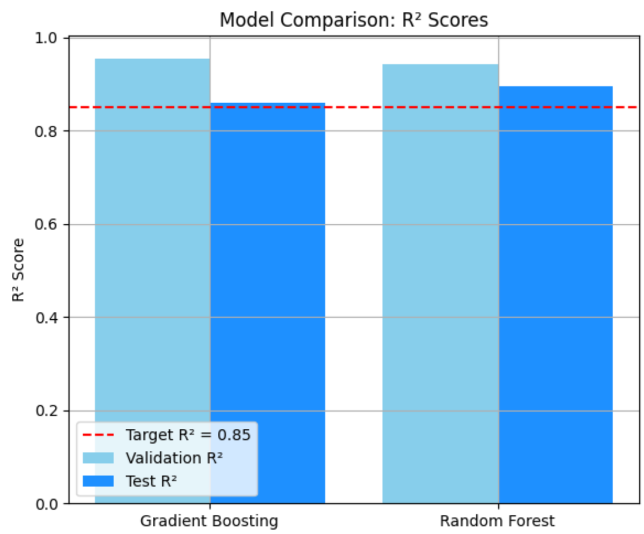

Fig. 8: Performance comparison of optimized Gradient Boosting and Random Forest models.

The final results demonstrate that both models significantly improved after hyperparameter tuning. The best validation and test performance scores achieved were:

- Gradient Boosting Performance: Validation R^2: 0.9552, Test R^2: 0.8587
- Random Forest Performance: Validation R^2: 0.9415, Test R^2: 0.8954

The results indicate that the Random Forest model outperformed Gradient Boosting on the test set by achieving a higher test R^2 score, which shows it was better at generalizing the estimation of energy consumption in unseen data.

To further evaluate the performance of the trained models, scatter plots were generated to compare the actual versus predicted EC values on the test set. Figure 9 presents the results for both Gradient Boosting and Random Forest models.

In each plot, the x-axis represents the actual EC values, while the y-axis represents the corresponding predicted values from each model. The dashed red line represents the ideal case where predictions perfectly match the actual values. Overall, the scatter plots confirm that the Random Forest model outperforms Gradient Boosting.

Lastly, the feature importance scores from the trained Random Forest model were analyzed to gain insights into which features had the most influence on energy consumption predictions. Figure 10 presents the feature importance ranking.

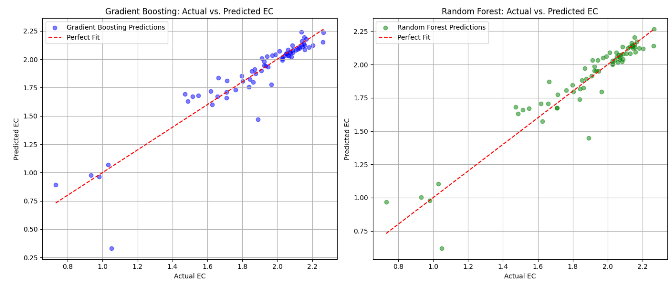

Fig. 9: Scatter plots of actual vs. predicted EC for Gradient Boosting and Random Forest models.

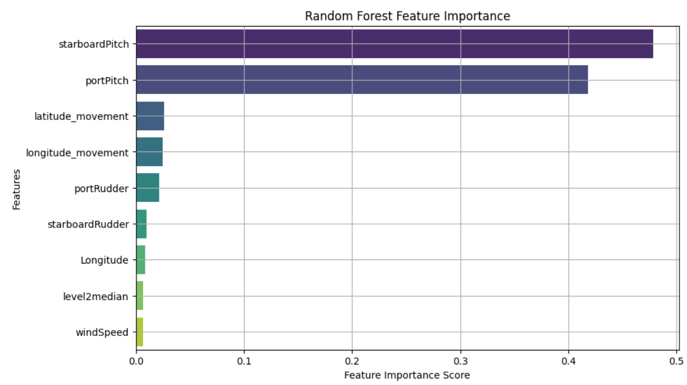

Fig. 10: Feature importance scores from the Random Forest model.

The results indicate that starboardPitch and portPitch were the most significant predictors, with both contributing nearly 50% to the model's decision-making process. Other important features include latitude_movement and longitude_movement, which represent the ship's displacement over time. Additionally, portRudder and starboardRudder showed a small impact on the predictions. Less influential features included Longitude, level2median, and windSpeed, indicating that their effect on energy consumption was minimal in comparison to the primary predictors.

V. Conclusion

This project examined data from multiple sensors installed on a ship, involving extensive data preprocessing, feature engineering, and feature selection. The goal was to predict the ship's energy consumption on an hourly-basis using machine learning models. After optimizing the models, a Random Forest model with tuned hyperparameters achieved an R^2 score of 90%, demonstrating strong predictive performance. The most influential features affecting EC were port and starboard pitches, rudder angles, and the ship's coordinational movements.

Several scientific bottlenecks were encountered throughout the process. The lack of domain knowledge made it difficult to interpret certain features and their impact on fuel consumption. To address this, exploratory data analysis was conducted, and feature importance techniques were used to determine which variables contributed the most to EC predictions.

Another challenge was the large dataset size, which made merging all data without exceeding memory limits difficult. This was mitigated by processing each dataframe separately before merging them.

Highly correlated features were also an issue, as they could introduce multicollinearity and negatively impact model performance. To resolve this, correlation analysis was performed, and redundant features were removed, ensuring that only the most informative variables were retained.

Timestamps were not evenly spaced, and it required careful handling of temporal data. Instead of assuming uniform time intervals, the dataset was aggregated into hourly intervals. Most features were averaged per hour, while others, such as distance traveled, were computed using the cumulative sum of time differences between consecutive rows to ensure accurate calculations.

Initially, six models were tested, with Gradient Boosting and Random Forest demonstrating the best performance. After hyperparameter tuning, the Random Forest model outperformed Gradient Boosting, achieving higher accuracy and better generalization to unseen data. The superior performance of Random Forest can be attributed to its ability to handle nonlinear relationships, robustness against overfitting, and feature importance weighting, which helped in making better predictions based on the most relevant variables.

For future improvements, several approaches could be explored. Increasing the dataset size by collecting additional voyage data could help improve model robustness. Feature engineering could be further refined by incorporating external factors such as sea conditions, ship load weight, and weather data. Additionally, experimenting with ensemble learning techniques or deep learning models could further enhance predictive accuracy.

VI. Accessing the Project

The work for this project can be found in notebook. To replicate the results, all 20 CSV files should be uploaded to a personal Google Drive and the variable folder_path should be updated accordingly.