Energy Consumption Predictor for Ro-Ro vessels from sensor data

GALBOKKA HEWAGE ISHARA MADHAVI

I. INTRODUCTION

Maritime transport plays a crucial role in global trade and logistics, with energy consumption being a key factor in operational efficiency and sustainability. Maritime transport is responsible for a significant portion of global greenhouse gas emissions, and optimizing fuel consumption directly contributes to reducing carbon footprints. Additionally, with increasing fuel prices and stricter regulations on emissions, ship operators are under pressure to improve energy efficiency.

Theoretically, energy consumption is computed using the fuel consumption rate, considering fuel density and volume flow rate. Real-world sensors used for fuel flow, speed, and environmental factors are prone to errors and inconsistencies. Factors like sensor drift, calibration issues, and missing data make direct calculations unreliable. Furthermore, the traditional approaches do not always consider external factors like wind speed, ocean currents, and wave height, which significantly impact a ship's fuel consumption. Also, the fact that ships experience frequent load changes, varying cargo weights, and different voyage conditions, which affect energy use, should not be avoided. By using historical data, machine learning models can capture complex relationships, adapt to changing conditions, and improve accuracy compared to static formulas. Additionally, models can be updated with new data, learning from past voyages to make smarter predictions in the future.

This study aims to use Machine Learning techniques to predict the energy consumption of the Danish Ro-Ro passenger ship using historical data recorded between February and April 2010, consisting of 246 voyages and over 1.6 million data records. The ship is equipped with multiple sensors, such as Doppler speed logs, gyrocompasses, GPS, fuel flow meters, rudder angle sensors, wind sensors, and propeller pitch sensors. These sensors provide valuable information on the ship's operational status, thus shedding light on pathways to use Machine Learning models for predicting energy consumption.

The goal is to develop robust machine learning models to predict energy consumption, aiding in fuel efficiency optimization and environmental impact reduction. The predictive models developed in this research can help ship operators make data-driven decisions to improve fuel efficiency and reduce emissions.

The rest of the paper is organized as follows; Section II presents the Data Processing steps followed to perform basic feature engineering and specific dimensionality reduction techniques used. Section III illustrates the Data Modelling which includes Data Split, Model Selection, Training and Performance Evaluations. The paper concludes with Section IV with final thoughts on the project and the conducted analysis.

II. DATA PROCESSING

The Data Processing step can be well explained in two phases as follows: Feature Selection, Window Interval Selection, Data Preparation and Feature Engineering.

1. *Feature Selection*

Selecting appropriate features is crucial for building an effective predictive model. Features were chosen based on their correlation with the target variable (i.e. Energy Consumption). A correlation heatmap was created to identify relationships among different features, and with the target variable.

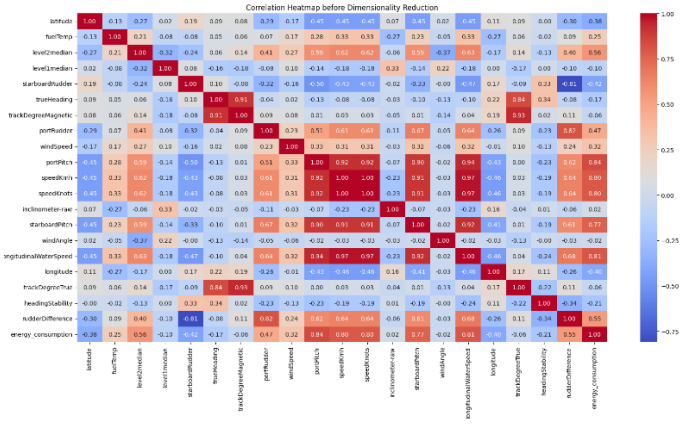

Figure 1. Correlation Heatmap before Dimensionality Reduction

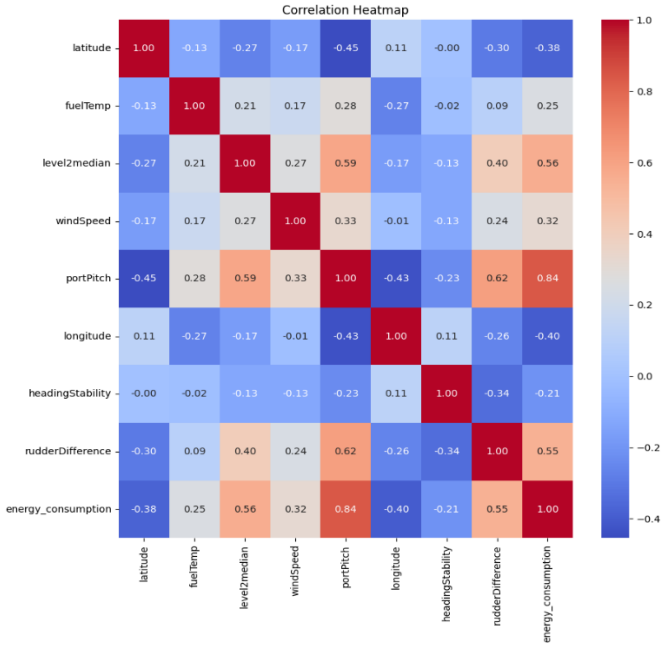

Figure 2. Correlation Heatmap after Dimensionality Reduction

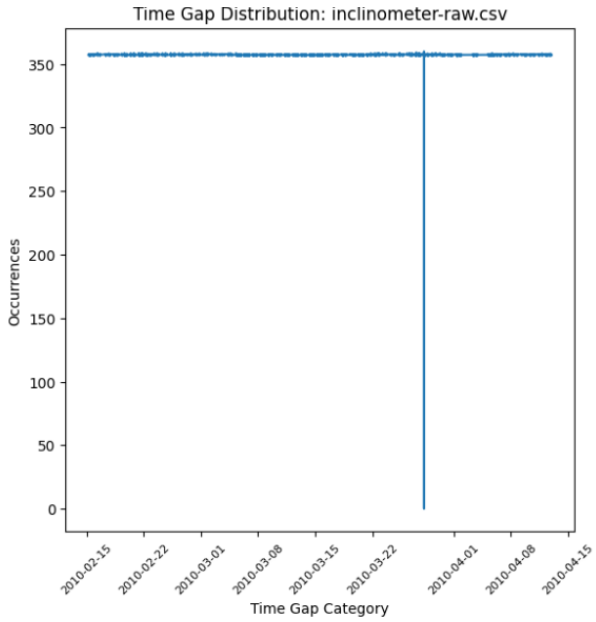

Figure 3. Distribution of inclinometer-raw values. The variation of values except for the outlier is negligible.

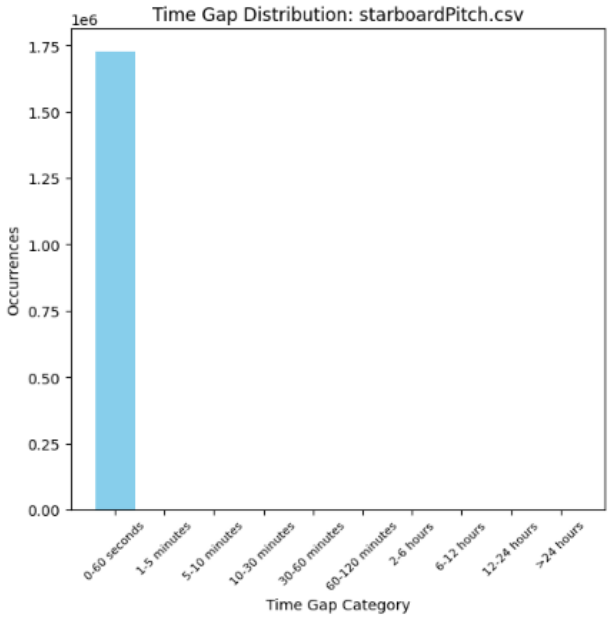

Figure 4. Distribution of consecutive time gaps in the starboardPitch dataset

From this analysis, redundant features which were highly correlated with each other were removed to prevent noise and overfitting, ensuring that only independent features with high correlations to the target variable contributed to the model. As illustrated in Figure 1, *portPitch* was the highest correlated feature to the target variable, and numerous other features such as *starboardPitch*, *speedKmh*, *speedKnots*, *longitudinalWaterSpeed* displayed correlation values higher than 0.90 with the *portPitch* feature. Therefore, it was decided to remove this additional feature set.

Additionally, features with minimal correlation to the target variable, such as the *inclinometer-raw* value, were excluded due to the negligible variation of inclinometer values as indicated clearly in Figure 3. Feature selection and dimensionality reduction helped improve model generalization and reduce computation time.

2. *Window Interval Selection*

To determine the optimal window interval for predicting energy consumption, an analysis of time gaps between consecutive timestamps was conducted. As illustrated in Figure 4, in every feature column, the majority of time gaps fell within the range of 0-60 seconds.

Metric	Value
Mean	2.84 seconds
Std. Deviation	243.8 seconds (~4 min)
Min	1.0005 seconds
25% - Q1	1.0006 seconds
50% - Q2	1.0006 seconds
75% - Q3	1.0007 seconds
Max	124,876 sec (~34.7 hrs)

Table 1. Statistics of Consecutive Time Gaps in starboardPitch data

Furthermore, as indicated in Table 1, the median (50th percentile) time difference is approximately 1 second, meaning that half of the recorded timestamps arrive within 1-second intervals. Additionally, the 25th and 75th percentiles are also around 1 second, confirming that the majority of the data points have minimal time gaps. However, the maximum gap observed is approximately 34.7 hours (124,876 seconds), indicating the presence of missing timestamps over extended periods. The high standard deviation (243.8 seconds ≈ 4 minutes) further suggests variability in time differences, primarily due to these occasional large gaps. Given that 75% of the data points have time gaps of about 1 second, choosing a 1-minute resampling window strikes a balance between reducing redundancy and preserving meaningful trends. Resampling at 1 second would generate excessive data points, leading to increased noise and computational overhead, while a 1-minute window smooths fluctuations without losing critical patterns in the dataset.

Based on this analysis, a **one-minute interval** was chosen for data aggregation. This decision ensures that meaningful patterns in energy consumption are retained while reducing unnecessary fluctuations caused by transient sensor noise.

3. *Data Preparation*

After visualizing the data distribution, performing a statistical analysis of data, and selecting a resampling window of 1 minute, below steps were performed, in order to curate the data for the training phase.

- Timestamps of data files were converted to a user-friendly format. The original values were in Windows File Time that represents the number of 100-nanosecond intervals since 1601-01-01. However, the conversion did not result in values in the desired date range (February 2010 to April 2010). Therefore, a minor adjustment was done in the timestamp conversion utility function.

- In the dataset, some large gaps between data points were observed with a maximum which was close to 3 days. If resampling was continued on these large gaps, it could result in a significant amount of artificial data inflation. To avoid this, the gaps between consecutive timestamps higher than a 1-minute interval were skipped from the resampling process.

- Longitude and Latitude data contained suffixes "N" and "W", which required cleaning, and conversion to numerical values.

- Data visualization revealed missing records at certain timestamps. To avoid artificial data imputation, the average values were calculated over one-minute periods. Larger time gaps were not interpolated to prevent incorrect values.

- Different files had varying start and end timestamps, leading to *NaN* values upon merging. To create a consistent dataset, the latest start time and earliest end time were selected.

- The target variable, energy consumption (*EC*), was computed as: ensuring calculations were made every minute.

$$EC = fuelDensity * fuleVolumeFlowRate * 60$$

- After calculating values for the target column, the *fuelDensity* and *fuelVolumeFlowRate* columns were dropped, since these features directly influenced the target.

- Data was standardized using Standard Scaler [1] to enhance model performance. Standardization helped models converge faster and improved their predictive capability.

4. Feature Engineering

Feature engineering plays a significant role in improving predictive performance, as derived features can capture hidden relationships between variables. Therefore, additional features were engineered based on domain knowledge and tested for correlation with the target variable.

For example, the ***headingStability*** captures the deviation between actual heading and intended track: A ship's true heading represents where the bow is pointing, while track degree true represents the actual course over ground. The difference between these two values reflects the ship's drift due to ocean currents, wind forces, and steering adjustments. rate of change of rudder angles provided insights into maneuvering patterns that influenced fuel consumption.

Additionally, ***rudderDifference*** captures asymmetric steering efforts: Ideally, both rudders should be synchronized for efficient navigation. A large difference suggests uneven steering efforts, which can lead to increased drag and fuel consumption.

Notably, *headingStability* and *rudderDifference* exhibited higher correlation values than their individual components, making them valuable additions to the feature set. Therefore, these features were also utilized for the model training process.

III. DATA MODELLING

After completing the data preprocessing step, two separate regression models were trained, evaluated and tested. The process that led to data modelling is as follows.

1. Data Split

The dataset was split into training and testing sets by 80:20 proportion.

2. Model Selection

Two regression models were selected to predict energy consumption:

1. **Random Forest Regression** - An ensemble learning method that combines multiple decision trees to improve predictive performance and reduce overfitting.

2. **K-Nearest Neighbors (KNN) Regression** - A distance-based method that predicts values based on the similarity of past observations.

3. Model Training

The training process was carried out by performing 5-fold Cross Validation on the training set. During the Cross-Validation phase, the R-Squared Scores were measured. Since the target variable, energy consumption displayed a skewed distribution as illustrated in Figure 5, Cross Validation was essential in measuring the Model Performance in a better way such that it provided useful insights into how the model would generalize for unseen data.

- A **5-fold cross-validation** approach was employed to ensure generalization. This technique divides the dataset into five subsets, training the model on four subsets while testing on the remaining one, iteratively improving performance.

- The dataset was split into **80% training and 20% testing** to assess generalization capabilities.

- The models were evaluated using the **R-squared metric (R^2)**, which measures the proportion of variance explained by the model. Higher values indicate better performance. Both models achieved **over 85% accuracy** on test data, demonstrating strong predictive capabilities.

- Hyperparameter tuning was conducted using **GridSearchCV**, optimizing the model parameters.

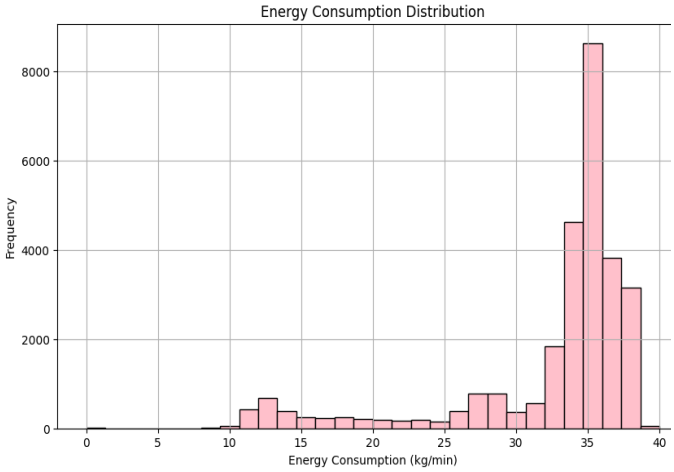

Figure 5. Target Variable (Energy Consumption) skewed distribution

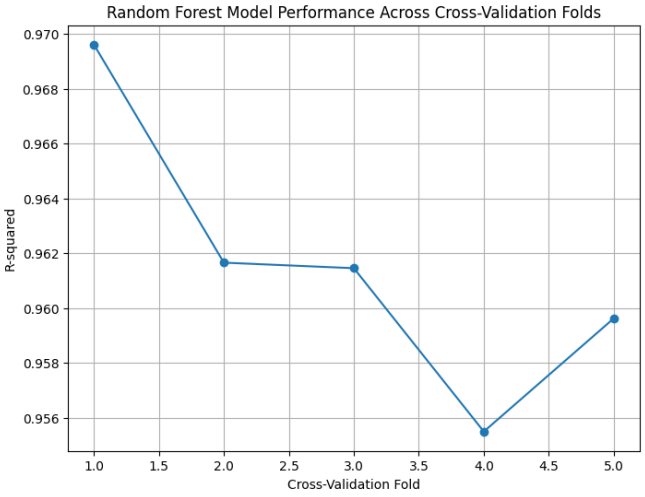

Figure 6. Random Forest Model R-Squared Score across 5 folds of cross validation during training

4. *Model Comparison*

Both the models achieved R-Squared scores exceeding 85%. In both, the R-Squared scores reported on the test data split was significantly closer to the respective average cross validation scores. This implied that the model was not overfitting and generalized well for unseen data points. Random Forest Regression displayed the highest average cross validation R-Squared score (96%) outperforming the KNN Regression model (91%) as indicated in Figure 6 and Figure 7 .

The K-Nearest Neighbors (KNN) regression model performed slightly lower compared to the Random Forest Regressor due to its sensitivity to high-dimensional data and its reliance on local patterns. KNN determines predictions based on the average of the nearest neighbors, making it highly dependent on the density and distribution of training points. In complex, high-dimensional datasets like energy consumption prediction, feature interactions and non-linear relationships become significant, which KNN struggles to capture effectively.

Additionally, KNN's performance is influenced by noisy data points and irrelevant features, as it lacks an internal mechanism for feature selection. On the other hand, Random Forest leverages multiple decision trees to learn intricate patterns, reduces variance through ensemble averaging, and provides better generalization. Its ability to rank feature importance and handle noise effectively contributed to its superior performance. In summary, the Random Forest Regression model performed better than the KNN-Regression model for the following reasons.

- **Better handling of non-linear relationships** between features and energy consumption.
- **Robustness against noise and missing data**, leveraging decision tree ensembles.
- **Feature importance analysis**, allowing better interpretability of the influencing factors.

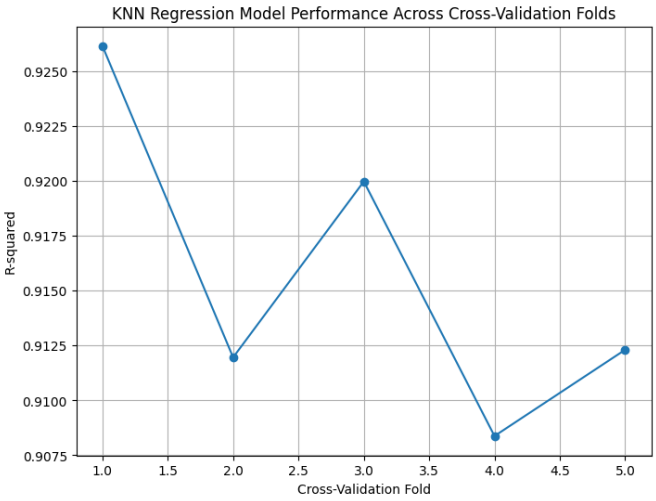

Figure 7. KNN Regression Model R-Squared Score across 5 folds of cross validation during training

5. *Performance Improvements*

Although both the models have shown relatively good performance based on the selected approach, as a further model enhancement step, Hyperparameter Optimization was selected and executed using the *GridSearchCV* [2] method. This method measures the cross-validation scores accumulated during the training process for a pre-defined grid of hyperparameters.

Hyperparameter Optimization increased the performance of the KNN-Regressor model by 2% by finding the best set of hyperparameters that achieved the highest average cross validation scores.

However, in the Random Forest Regressor model, the hyperparameter tuning did not significantly improve the model accuracy. The reason hyperparameter optimization did not significantly improve the accuracy of the Random Forest Regressor is that the default parameters of the model were already well-suited for the dataset. Random Forest is inherently robust due to its ensemble nature, which averages

multiple decision trees, reducing overfitting and variance. The default number of trees (often 100 in most implementations) is typically sufficient to stabilize predictions. If the model was already capturing most of the variance in the data, fine-tuning parameters like the number of estimators, maximum depth, or minimum samples per split might not yield noticeable improvements. This indicates that the baseline model was already close to optimal, and further tuning did not significantly alter its predictive power.

One significant fact to state is that the models did not achieve the reported R-Squared values in the first run. Therefore, as a post-processing step, the selected features were revisited, where more feature engineering was performed, as a result of which the final engineered feature set was selected, that included the *rudderDifference* and *headingStability*. This step was crucial for the improvement of the model accuracy.

IV. CONCLUSION

This study successfully developed machine learning models to predict the energy consumption of the MS Smyril vessel [3]. Random Forest Regression outperformed KNN Regression due to its robustness and ability to capture complex relationships.

Key scientific challenges encountered during the project, and the steps to address the bottlenecks are as follows.

- **Determining the optimal window interval**, requiring thorough analysis of data distribution and patterns. It required histogram analysis on time gaps to determine the dominant frequency. A higher resampling rate could result in loss of significant details and latent trends in data, while a lower resampling rate could be noisy and hence leads to overfitting. To overcome this challenge, a statistical analysis of the data was conducted, and by analyzing the mean, standard deviation, min and max values of the consecutive time gaps, an informative decision was made to select the optimal window.

- **Addressing data imbalance**, as certain time periods contained significantly more data records than others (i.e. the energy consumption variation in Figure 5 illustrated a skewed distribution). To address this bottleneck, instead of training and testing on a single static split (which might be overfit to dense data periods), time-series cross-validation splits the data sequentially, ensuring that both high-density and low-density periods are incorporated into different folds.

- **Handling timestamp conversions** required careful observation of the data present on the timestamp column. Although the timestamps were in Windows File Time, which is measured in 100-nanosecond intervals since January 1, 1601, the conversion did not result in the time range mentioned in the original problem. Therefore, it was required to be adjusted to 2010, by calibrating the start year of Windows File Time.

- **Achieving the desired accuracy target** was also a challenge. The expected accuracy was not obtained during the initial execution. Continuous experimentation on feature engineering played a pivotal role in improving model accuracy, highlighting the importance of domain knowledge in maritime energy analytics.

Future work could explore deep learning approaches and additional sensor integration to further enhance prediction accuracy. Additionally, incorporating weather conditions and sea state parameters could further refine energy consumption models for more accurate real-world applications. Also, there exists a vast amount of regression models that can be used to address the problem at hand, Therefore, more experiments can be conducted in future for improved performance and accuracy.

REFERENCES

[1] Standardscaler scikit. Available at: https://scikit-learn.org/dev/modules/generated/sklearn.preprocessing.StandardScaler.html (Accessed: 15 October 2024).

[2] GRIDSEARCHCV scikit. Available at: https://scikit-learn.org/dev/modules/generated/sklearn.model_selection.GridSearchCV.html (Accessed: 15 October 2024).

[3] https://www.marinetraffic.com/en/ais/details/ships/shipid:181927/mmsi:231300000/imo:9275218/vessel:SMYRIL