



Master's Thesis

Information Technology and Embedded & Intelligent Systems, 120 Credits

Data Driven Energy Efficiency of Ships.

Machine Learning and Data Science, 30 credits

Halmstad 2022-06-22

Tarik Taspinar

Abstract

Decreasing the fuel consumption and thus greenhouse gas emissions of vessels has emerged as a critical topic for both ship operators and policy makers in recent years. The speed of vessels has long been recognized to have highest impact on fuel consumption. The solution suggestions like “speed optimization” and “speed reduction” are ongoing discussion topics for International Maritime Organization. The aim of this study are to develop a speed optimization model using time-constrained genetic algorithms (GA). Subsequent to this, this paper also presents the application of machine learning (ML) regression methods in setting up a model with the aim of predicting the fuel consumption of vessels. Local outlier factor algorithm is used to eliminate outlier in prediction features. In boosting and tree-based regression prediction methods, the overfitting problem is observed after hyperparameter tuning. Early stopping technique is applied for overfitted models.

In this study, speed is also found as the most important feature for fuel consumption prediction models. On the other hand, GA evaluation results showed that random modifications in default speed profile can increase GA performance and thus fuel savings more than constant speed limits during voyages. The results of GA also indicate that using high crossover rates and low mutations rates can increase fuel saving.

Further research is recommended to include fuel and bunker prices to determine more accurate fuel efficiency.

Keywords: Local outlier factor, k-nearest neighbors, random forest, gradient boosting, support vector machines, ensemble learning, ship speed optimization, genetic algorithm, DEAP, HyperOpt.

Acknowledgements

I would like to express my sincere gratitude to thesis supervisors Sławomir Nowaczyk and Abdallah Alabdallah for thesis constant support, guidance and motivation throughout the work. I would like to thanks to also Johannes Huffmeier and other members of RISE for being a source of inspiration and for sharing all the dataset of this research.

Table of contents

List of Figures	1
List of Tables	3
1 Introduction.....	5
1.1 Background	5
1.2 Research Problem	5
1.3 Literature Review	6
2 Methodology.....	9
2.1 Linear Methods for Regressions	9
2.1.1 Linear Regression Model	9
2.1.1.1 Assumptions of Linear Regression	9
2.2 Nonlinear Methods for Regressions.....	10
2.2.1 Hyperparameter Optimization Techniques	10
2.2.1.1 Grid Search	10
2.2.1.2 Random Search	10
2.2.1.3 Bayesian Optimization.....	10
2.2.2 Support Vector Regression	10
2.2.3 K-Nearest Neighbors.....	11
2.2.4 Random Forest Regression	11
2.2.5 Adaptive Boosting (ADABOOST).....	11
2.2.6 Gradient Boosting Regressor	11
2.2.7 Extreme Gradient Boosting (XGBoost).....	11
2.2.8 Voting Regressor.....	11
2.2.9 Stacking Regressor.....	11
2.3 Speed Optimization.....	12
2.4 Proposed Genetic Algorithm Model	12
2.5 Genetic Algorithm Model Implementation Steps	15
2.5.1 Individual initialization	15
2.5.2 Population initialization	17
2.5.3 Calculation of fitness values	17
2.5.4 Selection.....	17

2.5.5	Crossover	17
2.5.6	Mutation	18
2.6	Genetic Algorithm Parameters	18
2.6.1	Mutation and Crossover Probabilities	18
2.6.2	Population Size	18
2.6.3	Top/Remaining Speeds	18
3	Data Acquisition and Preprocessing	19
3.1	Data Source	19
3.2	Data Preprocessing	23
3.2.1	Data Cleaning	23
3.2.2	Resampling	24
3.2.3	Outlier (Anomaly) Detection	25
3.2.3.1	Local Outlier Factor Algorithm	30
3.3	Feature Selection	32
4	Modelling Evaluation and Results	34
4.1	Performance and Validation of Estimation Models	34
4.2	Testing the Assumptions of Linear Regression	34
4.3	Model Performance Evaluation	41
4.4	Parameter Tuning of Genetic Algorithm	47
4.4.1	Random Initialization	48
4.4.1.1	Parameter Tuning Results	49
4.4.2	Manual Initialization	53
4.4.2.1	Parameter Tuning and Delaying Results	54
5	Conclusion And Future Work	57
5.1	Conclusion	57
5.2	Future Work	57
6	Appendices	58
6.1	Appendix A: Method: Scikit-learn LocalOutlierFactor	58
7	References	59

List of Figures

Figure 1 Flowchart of the speed optimization problem	14
Figure 2 Actual speed profile of the vessel.....	16
Figure 3 A random or manual modification for initializing individuals.....	17
Figure 4 Fuel consumption raw data and voyage time window.....	21
Figure 5 Selected features for regression analysis-1	21
Figure 6 Selected features for regression analysis-2.....	22
Figure 7 Selected features for regression analysis-3.....	22
Figure 8 Selected features for regression analysis-4.....	23
Figure 9 Selected features for regression analysis-5.....	23
Figure 10 30 seconds down-sampling of fuel consumption raw data.....	25
Figure 11 Standard deviations in a normal distribution.....	26
Figure 12 A general box-plot representation of IQR	26
Figure 13 Fuel consumption data distribution and IQR.....	27
Figure 14 Histograms and IQRs of regression features	29
Figure 15 Reachability distances of k-neighbors [31]	30
Figure 16 LOF method outlier detection results on various combinations of k-neighbors and contamination parameter values.....	32
Figure 17 Extreme correlation coefficients of X1 and X2 scatters plots.....	33
Figure 18 Pearson correlation coefficients between features.....	33
Figure 19 Scatter plots of selected features versus fuel consumption included linear regression lines of best fit.....	35
Figure 20 Prediction and residuals of linear regression (a) Prediction vs observed (b) Normal distribution of residuals (c) Normalized residuals versus predictions (d) Trend line of quantiles.....	37
Figure 21 Test error versus neighbor numbers	38
Figure 22 Three depth of tuned random forest regressor on training dataset	38
Figure 23 RF-Hyperopt trial results for training dataset.....	39
Figure 24 Learning curves of LR and RF models.....	43
Figure 25 Learning curves of SVR and KNN models	44
Figure 26 Learning curves of ADABOOST and GBR models.....	45
Figure 27 Learning curves of XGBoost, VR and SR models	46
Figure 28 Predicted versus measured fuel consumption (kg/h) of tested top 4 models	47
Figure 29 Parameter tuning results of random approach	49
Figure 30 Speed profiles of best fits in random approach step-1, a) crossovers and b) mutations	50

Figure 31 Speed profiles of best fits in random approach step-2, mutation standard deviations.....	51
Figure 32 Speed profiles of best fits in random approach step-4, changing speed limit (a) and voyage speed for remaining distance (b).....	52
Figure 33 Fuel consumption results for 1 hour (a) and 2 hours (b) ETA delays	53
Figure 34 Parameter tuning results of manual approach.....	55

List of Tables

Table 1 Notations of the speed optimization model.....	13
Table 2 The implementation steps of GA	15
Table 3 Oil Tanker Specifications [27].....	19
Table 4 Selected parameters for regression analysis.....	20
Table 5 Missing column values in percentage	24
Table 6 Fuel consumption IQR method outliers ($\sigma = 2$).....	27
Table 7 Model tuning hyperparameters and running times	39
Table 8 Performance result of models for the training dataset	42
Table 9 Performance results of models for test dataset.....	42
Table 10 Tuning parameters in manual and random initializations	47
Table 11 Control parameters and constants in randomly initialized GA.....	48
Table 12 Control parameters and constant in manually initialized GA	54

ACRONYMS

ANN	Artificial Neural Network
ADA	Adaptive Boosting
EEDI	Energy Efficiency Design Index
ETA	Expected Time of Arrival
IMO	International Maritime Organization
IQR	Inter Quartile Range
GA	Genetic Algorithm
GB	Gradient Boosting
KNN	K-Nearest Neighbors
LR	Linear Regression
LOF	Local Outlier Factor
ML	Machine Learning
RF	Random Forest
RMSE	Root Mean Square Error
SEEMP	Ship Energy Efficiency Management Plan
SR	Stacking Regressor
SVR	Support Vector Regressor
VR	Voting Regressor
XGB	Extreme Gradient Boosting

1 Introduction

1.1 Background

Greenhouse gas emissions (GHG) in maritime transport are one of the main contributors of global warming. Moreover, shipping GHG emissions are projected to increase up to 130% of 2008 values by 2050 [1]. International Maritime Organization (IMO) aims to decrease the carbon intensity of shipping through implementation of the energy efficiency design index (EEDI) and so GHG emissions in the initial strategy of IMO. EEDI requires a minimum fuel consumption per tonne-mile for different ship types and size segments. Besides, The Ship Energy Efficiency Management Plan (SEEMP) is an operational measure that enables ship operators to increase fuel efficiency by applying any operational changes such as new voyage planning, weather routing, delay to ports, optimizations of speed, shaft power, trim, ballast, propeller and waste heat recovery systems etc. [2]. These operational changes require a mandatory data collection system of technical, operational and environmental parameters in maritime transporting [3]. For those reasons, using a fuel consumption data collection system started to be mandatory for ships over 5,000 gross tonnages in 2018. Furthermore, the further purposes of data collection system are following by data analysis and decision-making systems if required in future [4].

The new regulations regarding speed restrictions are discussed and new speed reductions are proposed by IMO that at certain times, the ship industry reduces the speed below design speed of vessels in order to reduce fuel consumption accordingly emissions and voyage costs. As a rule of thumb, if the ship speed decreases 10%, the corresponding engine power and the total required energy for the voyage decrease by 27% and 19%, respectively [5].

But the speed reduction discussions are still ongoing. It is still unclear the regulation to apply whether to the maximum speed limits or maximum average speeds. In addition, there exist many oppositions from the industry side to certain speed reductions for each trip. For example, Stena claims that it is not possible to reduce speed by 20% for each trip in high seasons. Also, Terntank thinks it will just penalize the building of new modern ships. Lastly, Swedish Oriented Line opposes speed reduction because they claim that more ships will be required to deliver the same amount of goods [6].

Research Problem

Contrary to popular belief, some ship operators claim that the fuel consumption can increase even if the speed is lowered due to inefficient

operation of engines [6]. Moreover, the studies also indicate that an optimum speed profile exists for every voyage, and deviations from optimum speed can cause more fuel consumption [6-8]. Slow steaming can be quite overestimated for fuel consumption savings. To support this view, ship companies from the industry describe how they found an optimal speed when they intended to decrease speed due to time chartering or port delaying limitations. Because the environmental conditions, traffic in ports, bunker prices, and many more parameters are affecting delays for shipping and speed is adjusting in real time. Therefore, in this study, the ship speed is optimized to minimize fuel consumption for given certain delays by ship operator. However, the optimum ship speed profiles are unknown and need to solve optimization problems. So, a genetic algorithm is used to find the best speed profile by random modifications on default speed profiles. On the other hand, to predict fuel consumption with a corresponding speed, a separate fuel consumption prediction model was investigated by applying machine learning models. As a consequence, a fuel consumption model developed for using as the objective function to minimize fuel consumption in genetic algorithm.

Besides, raw data has been recorded by a partner company of the stockholder. So, in this thesis, exploratory, predictive data analysis, and speed optimization are applied.

1.2 Literature Review

Energy efficiency studies in the shipping industry have become very common in literature. UCL Energy Institute (2016) [9] published an assessment of shipping operational efficiency of different fleets using satellite AIS data. The analysis of AIS data results showed that 10-15% decrease in the average speed of vessels could improve operational efficiency by approximately 10% for bulk fleets and up to 30% for container fleets. Additionally, the findings in the study support that due to the wide spread of energy efficiency results between ships in many fleets, consideration of individual speed decrease and energy efficiency of ships can improve the average energy efficiency of fleets.

However, over time, the opposite view has emerged. Karl et al. (2020) [6] discussed the consequences of speed reductions for ships in Swedish business. The calculation results showed that fuel consumption could also increase at lower speeds. Moreover, one of the key findings was the least fuel consumption can change according to the optimum speed for every cargo load level. Therefore, changing the speed up or down from that optimum point can cause more fuel consumption.

Indeed, earlier, the predominant opinion was assuming a cubic function of ship speed well explains ships' fuel consumption in case of missing historic data [10]. Thus, the design speed of ships has become the optimum speed for sailing also. In order to show the flexibility potential of speed around the

optimal point Adland et al. [10] took into account the noon report data of oil tankers. The study showed that the fuel consumption efficiency of speed reduction assessments is overestimated because using cubic methods for speed reduction results in more fuel savings than regression models. They also claim that the regression model used in the study was not capable of finding global optima for predictions even a non-linear model is proposed. Furthermore, the fuel consumption prediction accuracy of linear regression model was around 80% for Aframax and Suezmax tankers. On contrary, using machine learning boosting ended with lower accuracy results. But neural networks with five hidden layers ended with higher accuracy results.

So, fuel consumption prediction with higher accuracy and lower error rates is a prerequisite for optimization studies. Kim et al. [11] analyzed ANN or multiple linear regression (MLR) models tested for fuel consumption prediction using main engine RPM, speed over ground, wind speed, rudder angle, draught, trim, wetted surface area, and displacement parameters. Eventually, ANN and MLR models ended with 99% and 87% accuracy results, respectively.

A large number of variables affect fuel consumption in ship data. But some of the variables have only noise effect in model and do not contribute to the prediction accuracy. Furthermore, they can cause overfitting that the model learns unnecessary details for the outcome and negatively impacts the algorithm performance. So, variable selection is another important step in machine learning prediction models. Regarding this issue, Corradu et al. [12] applied Brute Force Method (BFM), which is the most accurate but shows computationally heavy performance, Lasso Regularization which has lower computational cost and Random Forest Method (RFM) which uses decision trees with permutation tests to select the features. Eventually, BF methods ended with better accuracy results. More precisely, the propeller pitch and ship speed parameters had more impact to fuel consumption prediction. On the contrary, the propeller speed is not among them due to being constant long time along the trip. Also, ship draft and shaft power were among the most important parameters for fuel consumption prediction. But, despite assuming to be relevant to fuel consumption, wind speed and its direction were not resulted having a serious impact on output.

In order to avoid overfitting due to overestimated feature selection, using non-parametric models could be a good option. Gkerekos et al. [13] used a bunch of parametric and non-parametric ML models together to find the best model performance. Ridge and Lasso regression, MLR, Support Vector V (SVR), ANN are used as parametric models and Decision Tree Regressor (DTR), Extra-Tree Regressor (ETR), K-Nearest Neighbors (KNN) and Random Forest Regressor (RFR) are used as non-parametric models. The assumption of the study was getting higher performance from non-parametric models due

to running without parameters assumptions in the beginning. Best accuracy results were obtained from randomized ETRs.

Fuel consumption prediction accuracy is significantly important to calculate fuel consumption in speed optimization model closest to real life scenario. So that speed optimization implements with the highest accuracy model between proposed machine learning models. Yang, Chen [14] implemented a genetic algorithm for speed optimization of an oil tanker. The whole route is divided into several stages and the speed is corrected considering wind waves and ocean currents. Maximum and minimum sailing speeds and expected time of arrivals are applied for genetic optimization constraints. Moreover, fuel consumption is considered as an objective function to minimize. Eventually, selecting correct speed for each segment resulted in 2.20% less fuel consumption.

Expected time of arrival can be changed by many unexpected events in maritime shipping. The waiting and delay in ports can be extra cost (delaying penalty) in liner shipping. Aydin, Lee [15] investigated sailing speed with variable ETA delays and bunker prices. Dynamic programming is used to determine sailing speed considering by uncertain port times. Because vessels can change the speed by checking next port congestion and bunkering prices at the same time. Furthermore, alternatively a deterministic method is also applied using expected values of random quantities in dynamic programming. Average sailing speeds resulted higher than dynamic model. Furthermore, the performance of deterministic model decreased as the delay penalty increased.

Expected time of arrival and fuel consumption conflicts with each other. Shortening the ETA causes more fuel consumption and vice versa. Helong, Xiao [16] developed a voyage optimization model using genetic algorithm with two different objective functions: Minimize fuel consumption and increase arrival punctuality. Planning routes heuristically under harsh sea conditions voluntarily choose low speed and helped fuel consumption saving up to 3.4% by keeping same ETAs. Also, this study has showed that increasing the genetic variety in genetic algorithm populations increases the success of objective function i.e., fuel consumption savings.

2 Methodology

The fuel consumption prediction is prerequisite function of speed optimization. In this section, linear and nonlinear regression models proposed for fuel consumption prediction are introduced. Moreover, the best parameters are searched by hyperparameter optimization for all models. In addition, voting and stacking ensemble methods are used to get best performance from regression models.

On the other hand, genetic algorithms are used for speed optimization. The best speed profile is searched by random processes in genetic algorithm. A random and manual approaches are developed to increase genetic variety.

Besides, data cleaning, resampling, outlier detection etc., preprocessing techniques are applied to raw data. Finally, a feature selection procedure is implemented for regression section.

2.1 Linear Methods for Regressions

2.1.1 Linear Regression Model

2.1.1.1 Assumptions of Linear Regression

Linear regression is a parametric analysis and it assumes that all the variables already have five certain features [17]. They are listed below.

- a) There are linear associations between dependent and independent variables. The magnitude of correlation determines the explainability and error of linear regression.
- b) There is no correlation between independent variables. Ideally, the magnitude of R values should be 0, but it is not a common situation in practice.
- c) There is no correlation between residual (error) terms. In other words, no autocorrelation exists.
- d) The error terms have constant variance. In other words, no heteroscedasticity exists.
- e) The error terms are normally distributed.

Before using linear regression models, the assumptions above should be proven by visual tests. For that reason, the visualizations belonging to all regression variables show how they are suitable or not for linear regression modeling.

2.2 Nonlinear Methods for Regressions

For this reason, Support Vector, K-Nearest Neighbors, Random Forest, AdaBoost, Gradient Boosting, and XGBoost regression models are implemented. Moreover, Grid Search, Randomized Search and Bayesian Search methods are used to tune their hyperparameters. Also, the training and test dataset performance results of models are compared in Table 5 and Table 6 to determine the best model for speed optimization.

The dataset is split into training and test subsets by an Scikit-learn data splitter method. The dataset has been shuffled before splitting to avoid overfitting and the test dataset was never used for model training.

2.2.1 Hyperparameter Optimization Techniques

2.2.1.1 Grid Search

Grid Search (GS) is a brute-force method that searches all the input space and also has a computational complexity of a cartesian product of k hyperparameter space with n possible values, $O(n^k)$ [18]. So, searching more possible parameter values with GS increases the computational cost exponentially. But, in this study, the GS is used for KNN, SVR, ADA, and GBR models.

2.2.1.2 Random Search

Instead of GS, Random Search (RS) does not search all the input space but uses only randomly selected values for parameters for the iteration number defined by the user [18]. Increasing the iterative numbers results in better scores for the model but also computational costs also increase. In this study, the RS is used for RF, XBG, and GBR models.

2.2.1.3 Bayesian Optimization

The main disadvantage of both GS and RS methods is evaluation without memory of previous evaluation results [18]. But Bayesian methods uses past evaluation results of hyperparameters for the next input values. In this study, one of the Bayesian Optimization techniques, Python HyperOpt library used to implement hyperparameter optimization. In this study, the RS was used for the RF model.

2.2.2 Support Vector Regression

Support Vector Regression (SVR) is an application of support vector machines (SVM) and also a popular tool for solving the fuel consumption regression problem of ships in maritime literature [19]. Kelleher, Mac Namee [20] identify SVR strengths that “they can be quickly trained, are not overly susceptible to overfitting, and work well for high-dimensional data.”

In this study, support vector regressor method of Scikit-learn support vector machines(sklearn.svm) module is used to implement the SVR model. In

addition, Scikit-learn ensemble module was also used to implement all the following ensemble models.

2.2.3 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a non-parametric model based on the classifying of points by using neighborhood and differ from other models which use past data in training [21]. KNN algorithms show poor performances in the conditions when the dataset are imbalanced or very large, because KNN can easily determine the target in favor of major group in dataset or the distance calculation cost increase enormously in large datasets [20].

In this study, the default parameters except the neighbor number of the Scikit-learn framework shown in Appendix-A are used to implement KNN algorithm. To avoid overfitting, the neighbor numbers can be tuned iteratively by checking RMSE values.

2.2.4 Random Forest Regression

Random forest (RF) is based on decision trees but also uses randomly selected samples in a dataset. Kelleher, Mac Namee [20] emphasize that “subspace sampling further encourages the diversity of the [decision] trees within the ensemble and has the advantage of reducing the training time for each tree”. Because individual decision trees can include high variance and overfitting, however, an averaging of errors by randomly selecting trees in random forest can decrease these errors [22].

2.2.5 Adaptive Boosting (ADABOOST)

ADABOOST improves the decision tree algorithm by using subsequent weak learners of the training dataset in the successive iterations [21]. So, in other words, it learns from previous mistakes.

2.2.6 Gradient Boosting Regressor

Gradient Boosting Regressor (GBR) also learns from previous weak learners. GBR does not change the weights of previous learners as ADABOOST but instead matches new predictor to previous residual errors [21].

2.2.7 Extreme Gradient Boosting (XGBOOST)

XGBOOST (XGB) is an ensemble method that uses a differentiable loss function and gradient descent optimization to minimize loss.

2.2.8 Voting Regressor

Voting regressor (VR) is another ensemble model that combines different ML regressors and averages the predicted values. Thus, the overfitting can be eliminated

2.2.9 Stacking Regressor

Stacking regressor (SR) is an ensemble method that all the regression models in stacking regression are trained individually. Then, the selected final

regressor fits based on the training outputs of other models. By this method, the overfitting can be avoided for the next model in chain.

2.3 Speed Optimization

Speed optimization is evaluated in order to find the best speed profile resulting minimum fuel consumption per tonne-mile. Because according to Ship Energy Efficiency Management Plan (SEEMP), less than optimum speed can cause higher fuel consumption [2, 7]. So optimum speed is not the lowest speed, and also, it is not constant during the voyage [2, 8]. But in order to identify an optimum speed profile, it requires solving an optimization problem as well. However, weather and sea environmental variables and market values such as bunker prices, port time windows, delays etc., have some inconsistencies in real life. On the other hand, the deterministic approaches for solving an optimization of speed and fuel consumption of ships ignore random events. Furthermore, they assume that randomly occurring environmental or technical failure events are known in advance [15]. In addition, in literature, deterministic methods such as convex or cubic functions have been well studied for speed optimization in recent years. But, the random variables have not been considered yet explicitly for the speed optimization problem [15]. For some aspects, many studies in the literature employ a stochastic term for some of the influential factors of fuel consumption, such as weather and sea conditions or engine parameters. However, they assumed that the random influential factors of fuel consumption follow a normal distribution [15]. But if we assume that many more influential factors exist in fuel consumption, using only probabilistic distributions for optimization problems will not produce effective solutions. So, using more random variables can help to get more realistic predictions for fuel consumption. In this study, genetic optimization algorithms are used to search optimum speed profiles for the least fuel consumption. The main reason for the selection of an evolutionary algorithm is to avoid converging in a local optimal point, on contrary finding the global optimal point of minimum fuel consumption.

Besides, the results of regression analysis in the previous section have shown that fuel consumption is mostly dependent on ship speed than other parameters. So that it is obvious that speed optimization can easily result in more fuel savings than other factors.

In this study, ETA delaying, maximum allowable speed and acceleration of speed to decrease hull stress are considered as speed optimization constraints. In addition, in order to support the fuel-saving when a gradual increase in speed during leaving from the port is allowed [2].

2.4 Proposed Genetic Algorithm Model

In this section, a speed optimization mathematical model is developed for a single route between two ports. The notations of the model are given in Table 1.

Table 1 Notations of the speed optimization model

Parameters	Description
ETA	The ETA at destination port (h)
d	Sailing distance (mile)
d^{new}	Sailing distance of modified speed (mile)
t	Voyage time (h)
T^R	Additional time for selected remaining speed (h)
V^R	Remaining speed (mile/h)
V^{top}	Top speed (mile/h)
F_{total}	Total fuel consumption normal speed (kg)
F^R	Fuel consumption rate of remaining speed (kg/h)

Based on the notations in Table 1, the formulations are shown below. The following equation represents the distance to reach of modified speed profile:

$$d^{new} = \sum_{i=0}^n V_i^{new} \times t \quad (1)$$

The gap between the distance of the modified speed profile and the normal distance will be closed with the selected remaining speed by the ship operator. The required extra time for the new speed profile to complete the trip represents in Equation 2.

$$T^R = (d - d^{new}) / V^R \quad (2)$$

The total fuel consumption shown in Equation 3 is the sum of the corresponding fuel consumptions of the prediction model using all the speeds until the voyage distance is reached.

$$\min \sum F_{total} + T^R \times F^R \quad (3)$$

Subject to

$$t + T^R < ETA \quad (4)$$

$$V^{new} = \begin{cases} V^{new}, & \text{if } V^{new} < V^{top} \\ V^{top}, & \text{otherwise.} \end{cases} \quad (5)$$

The objective function at Equation 3 is used to calculate the fitness values of individuals in the genetic optimization algorithm. The main purpose of GA is to find minimum fitness-valued individuals. Therefore, the speed profile uses the constraints in Equation 4 and 5 by ensuring that the ship arrival time will not be longer than ETA and the maximum allowed speed constraint used for individual modification stages ensures that no overspeed above it.

The workflow of the proposed speed optimization method is shown in Figure 1.

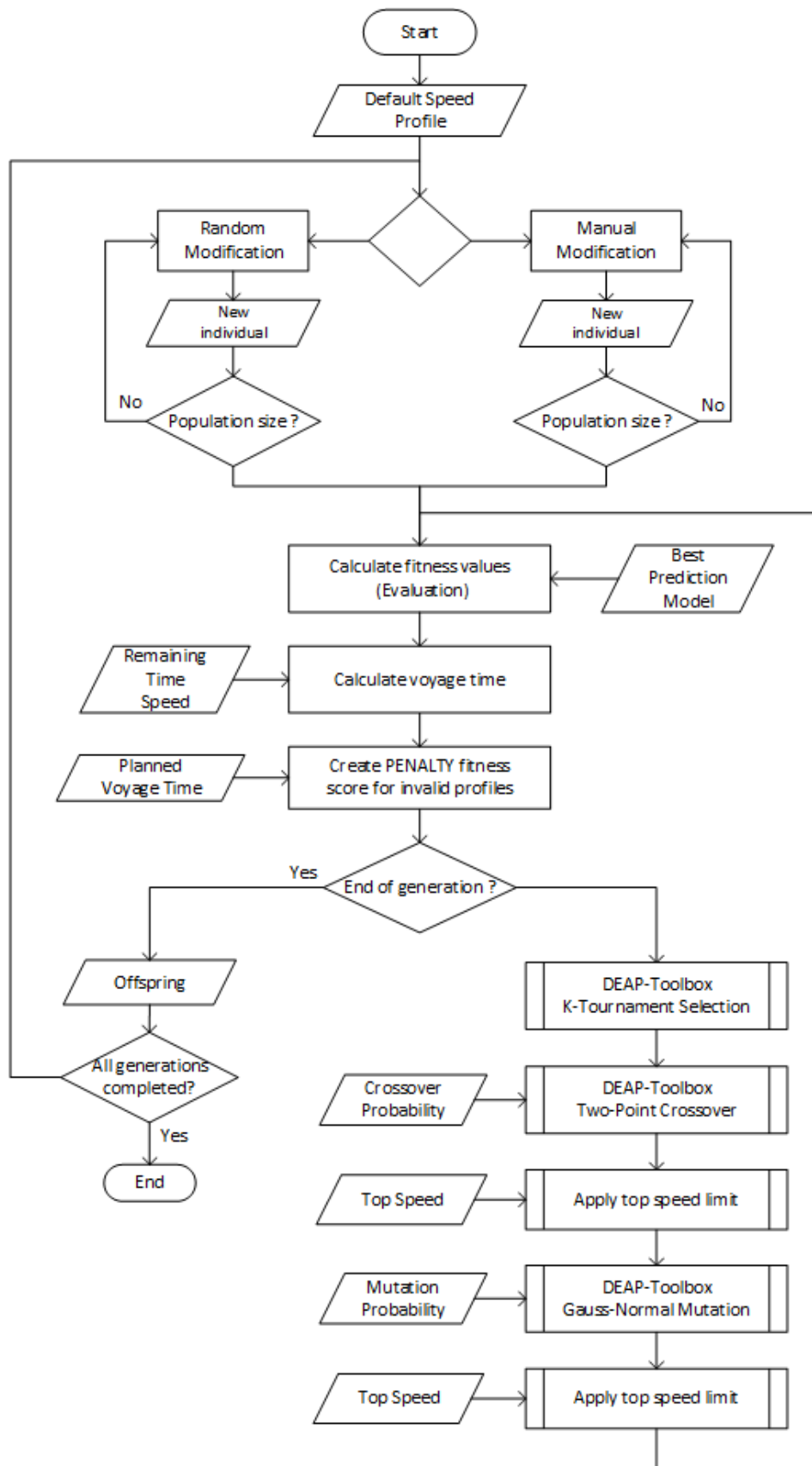


Figure 1 Flowchart of the speed optimization problem

2.5 Genetic Algorithm Model Implementation Steps

Apart from the mathematical implementation of optimization, the algorithm analysis is evaluated in this section.

In this study, the DEAP (Distributed Evolutionary Algorithms in Python) framework was used for the code implementation of all the genetic optimization algorithm methods. DEAP framework has mainly creator and toolbox module to create individual types such as float list of speed values and fitness/objective function to minimize. Also, the Toolbox module is used to implement selection, crossover and mutation operators by registering these functions to created population list of individuals. Furthermore, penalty and boundary checking decorator functions are registered to populations with the toolbox module.

The implemented GA optimization algorithm steps are listed in Table 2 and also explained further under separate headers.

Table 2 The implementation steps of GA

Steps	Description
1	Initialization of individuals using random or manual approach
2	Filling the population list with individuals
3	Evaluation of fuel consumption (fitness) values of all individuals
4	Selection
5	Crossover
6	Mutation

2.5.1 Individual initialization

Speed profiles are proposed as individuals of GA. Each individual evaluates by a fuel consumption score or fitness value. Each individual must have some differences in order to evaluate different fuel consumption scores. The actual speed profile of the vessel is presented in Figure 2.

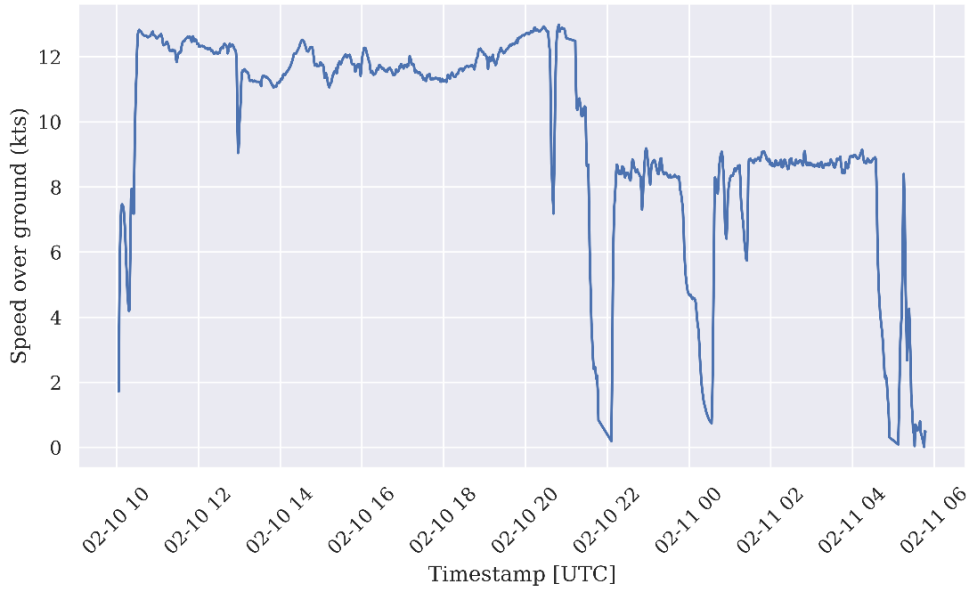


Figure 2 Actual speed profile of the vessel

In this study, the proposed speed optimization method searches for optimal sailing speed profiles by using evolutionary algorithms. But the initialization of these algorithms can cause convergence and robustness problems of speed optimization [16]. To overcome these problems, a special effort was exerted to create a high-level genetic diversity for the individuals.

For those reasons, a modification to the speed profile was applied to create different initial speed profiles for new individuals. Two different approaches are used to create initial speed profiles. The first one is making a random change in a randomly selected part of the original speed profile. Randomly selected amount of modification for speed reduction and its duration is applied to the original speed profiles. Random speed reductions can be assumed as variable elasticity of fuel consumption to speed variable. Likewise, constant reductions can be assumed as constant elasticity [10]. The elasticity around optimum speed is a research objective in this study. This random approach causes a good variety of speeds between individuals at the beginning of selection, crossover, and mutation stages. The second method is defining speed reduction and speed reduction time manually in order to compare the random approach. In Figure 3, the view of a random modification is represented.

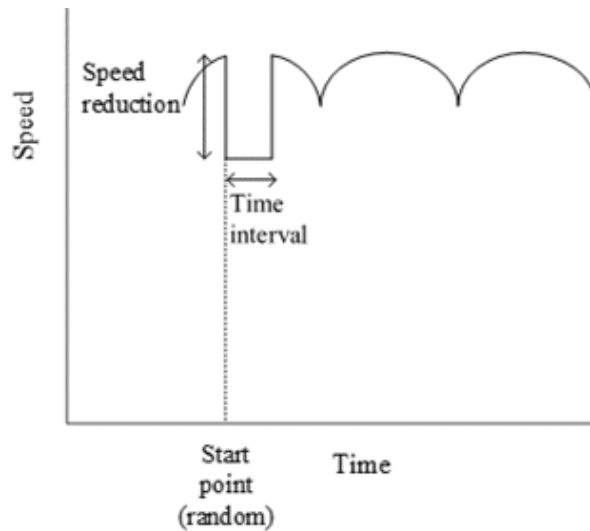


Figure 3 Speed profile modification for initializing the individuals

2.5.2 Population initialization

A population corresponds to N number of individuals. In this thesis, N number of new individuals are initialized by random or manual approach in every generation step.

2.5.3 Calculation of fitness values

Fitness is an objective function which has the goal of minimizing fuel consumption. Random or manual speed reductions, crossovers and mutations can cause a delay time or overtime for voyages. In evaluation algorithm, firstly, the new total voyage time is calculated by including the remaining speed. Then, a penalty fitness score assigns for the overtime profiles to inhibit the selection of them by GA.

The corresponding fuel consumption values are evaluated by best prediction model using new speed profiles also including the remaining speed. The total fuel consumption defines as the fitness score of individuals.

2.5.4 Selection

After all the individuals are evaluated in a population, they are ranked in with fitness values in increasing order. Because in this study, the purpose of the fitness function is to minimize fuel consumption.

The first step in a generation is the selection of offspring. K-Tournament selection method with 3 tournament size and k (1) time is used in this study. Every time, the best two individuals are selected for the crossover step.

2.5.5 Crossover

Best individuals are paired and mated according to crossover probability ratios [23]. Two-point crossover is used in this study and after each crossover step has been completed, top speed is checked and speed limit values are applied if any speed value exceeds the top speed limit.

2.5.6 Mutation

Mutation is another important step for increasing genetic variability. Mutations are required to continue searching for the best results as long as possible. Otherwise, the optimization wrongly assumes a local minimum or maximum point as global. In order to reduce the risk of wrong assumptions, using a certain amount of mutation rate is beneficial [16]. Randomly selected values in individuals mutated in gauss-normal distribution range defined with zero mean and a standard deviation. After the mutation step is completed, evaluation steps are run to create new fitness values of only modifications that occurred to individuals. All five steps above repeat until the end of all the generations completed. Finally, the mostly common generation size is selected as 99.

2.6 Genetic Algorithm Parameters

2.6.1 Mutation and Crossover Probabilities

Mutation and crossover parameters are major factors of genetic algorithms. In literature, some studies show that high crossover rates are the major factor for reproduction, but also high mutations can destroy the DNA, so they are used rarely [24]. In this study, two-point crossover was used after the selection steps of GA. Moreover, crossover rates were empirically investigated between 20 and 90 percent. Anyway, 80% [25] and 75% [26] crossover rates are commonly used values in literature. In addition, Gauss distribution with zero mean is used for changing the amount of speed values.

2.6.2 Population Size

The number of individuals or population size has a major impact on genetic algorithm performance. The problem space can be better analyzed by a genetic algorithm with higher population sizes, but it does not guarantee a better solution point because it depends on the problem itself [23]. In addition, optimization with a higher number of individuals consumes more computational time.

The optimum number of individuals depends on application complexity. Additionally, algorithm complexity and individual numbers have an effect on performance in the same direction. In literature, the recommended number of individuals is between 30 and 300. But also, the empirical methods are advised in order to find optimum value to show the best performance [24]. In this study, the adaptive sizing method is used to find optimum population size.

2.6.3 Top/Remaining Speeds

After defining ETA and remaining speed for delay time, GA calculates total fuel consumption as a sum of corresponding fuel consumption for remaining and modified speed curves by the selected fuel consumption prediction model.

3 Data Acquisition and Preprocessing

Data has been provided by a stockholder. Dataset has been recorded under various load and environment conditions for 69 months. Original data is recorded for six months [27]. The sampling rate of the original dataset is at every one second. In order to decrease calculation time and obtain better convergence of optimization algorithms, the sampling rate is decreased to 1 minute. But the signal information is still protected and avoided from aliasing.

3.1 Data Source

The data used in this thesis belongs to a specific oil tanker in Sweden provided by a stockholder. The specifications of this oil tanker are listed in Table 3. Tanker is specialized to carry heat cargo and therefore it has a larger ballast water tank. Besides, the draught difference between the back and front side of the ship defines as trim and it varies more when the ship is unloaded, but it also varies so small under ballast conditions [27].

Table 3 Oil Tanker Specifications [27]

Specification	Dimension	Value
Length over all	m	99
Design draught	m	5.7
Deadweight	t	4972
Year of build		2012
Cargo capacity	m^3	4300
Water ballast	m^2	2000
Main engine	kW	4000
Shaft generator	ekW	760
Service speed	knots	14.0

In between many environmental, operational and voyage data-logs belonged to this tanker, just 14 of them are selected in this thesis. Because only the variables which are mainly used in similar studies for fuel consumption prediction and speed optimization are considered during selection process. Fuel consumption data is the main predictive target in this thesis. But it can also be represented by the output power of main engines or propeller pitch. Even though the correlation between fuel consumption and main engines or propeller pitch is not a hundred percent; as a result, the fuel is consumed by only main engines via controlling propeller pitches. Therefore, to avoid losing

the prediction power of other parameters, main engines or propeller pitch data variables are excluded from this study.

Table 4 Selected parameters for regression analysis

Id	Parameter	Dimension
1	Speed over ground	knots
2	Consumption	kg/h
3	Ballast	ton
4	Diesel fuel	ton
5	Keel depth	m
6	Aux Generator	kW
7	Freight	ton
8	Water	ton
9	Angle of list	deg
10	Heading	deg
11	Trim	m
12	Tailwind	m/s
13	Wind speed	m/s
14	Wind direction	deg

Tanker ships carry heated cargo port to port. Voyage time varies depending on the distance between ports. In Figure 4, the observed fuel consumption is shown in 1-day range. It is noticed that the whole voyage starts and ends in the 24-hour interval. More precisely, it's between 9 AM to 6 AM the next day. In this thesis, fuel consumption analysis focused on more operational and environmental parameters in voyage time intervals than ship maintenance or maneuvering time intervals. Because engine parameters were not used in this thesis. For that reason, low fuel consumption values are excluded in this study. By using Figure 4, time before 9 AM and time after 6 AM are shown by red vertical lines excluded from raw data.

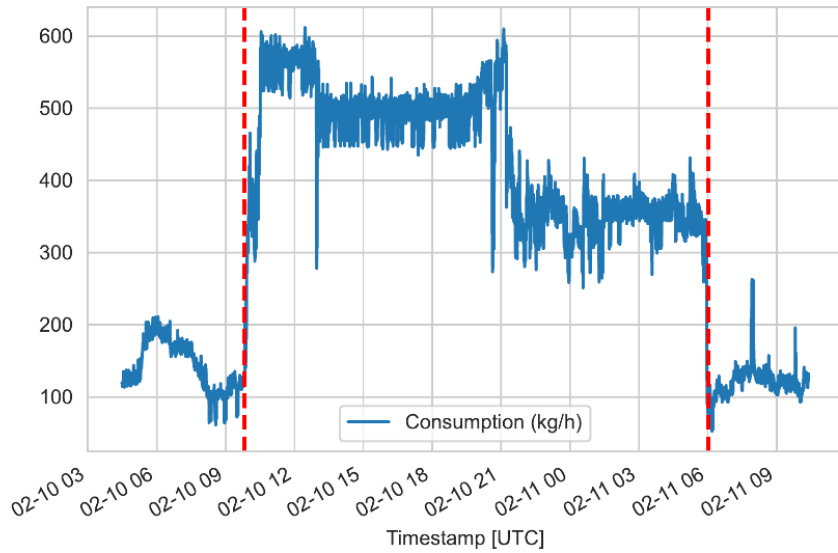


Figure 4 Fuel consumption raw data and voyage time window

After this data correction, in Figures 5-9, the selected variables for regression analysis in Table 2 are given in the time domain. The original sampling rate (1 second) is changed by 30 seconds averaged down sampling process applied to the whole dataset.

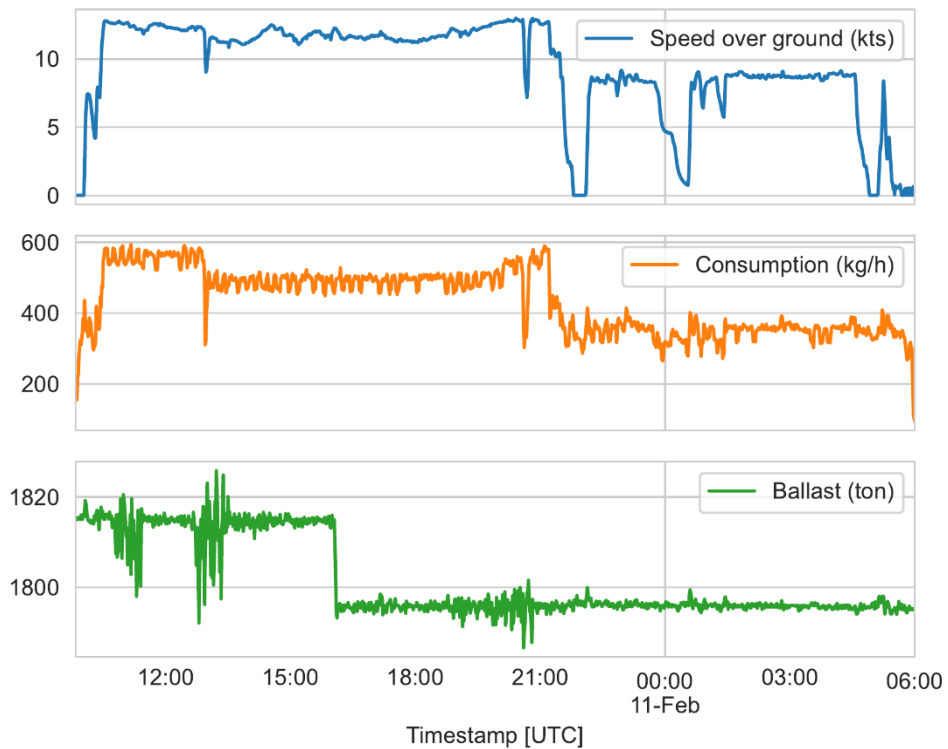


Figure 5 Selected features for regression analysis-1

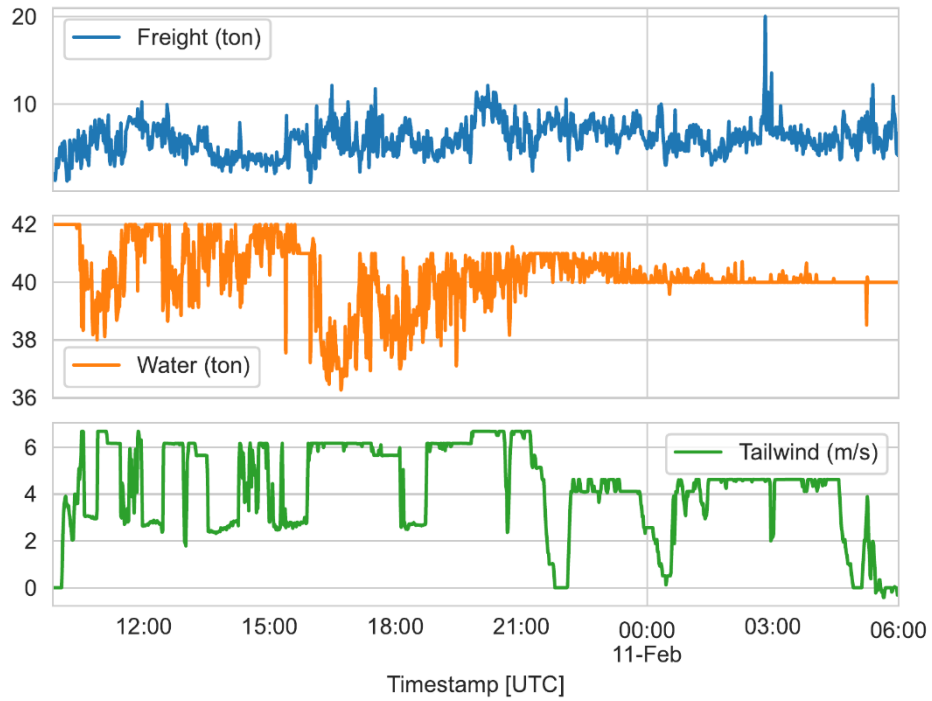


Figure 6 Selected features for regression analysis-2

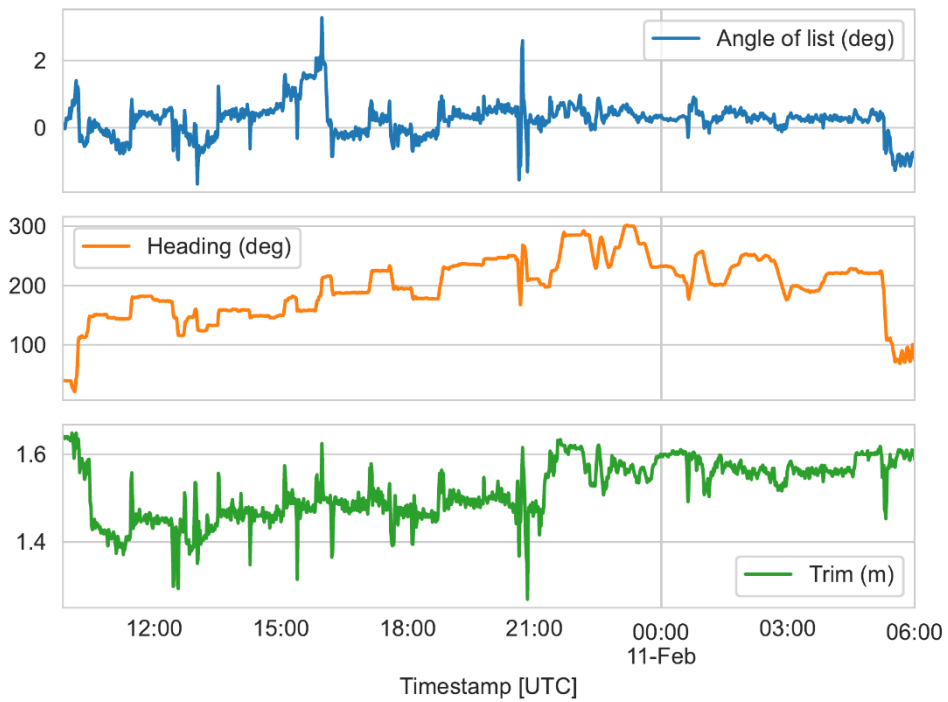


Figure 7 Selected features for regression analysis-3

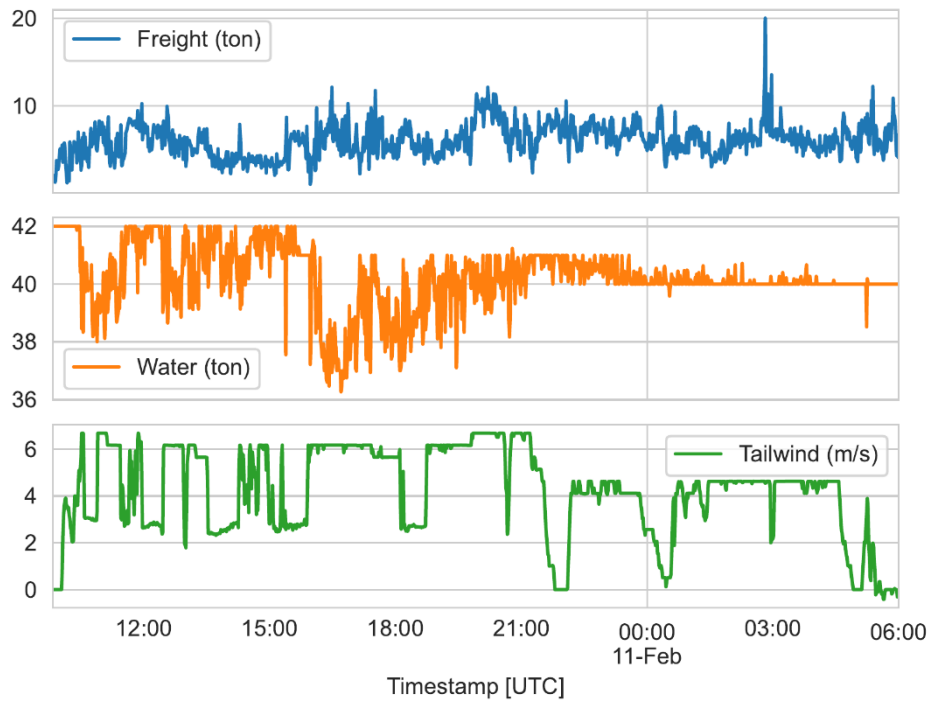


Figure 8 Selected features for regression analysis-4

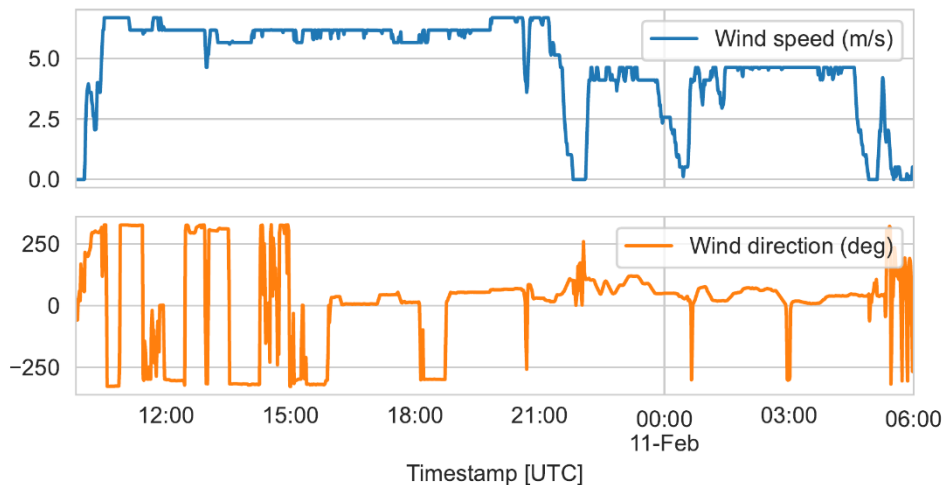


Figure 9 Selected features for regression analysis-5

3.2 Data Preprocessing

Before data analysis, raw data is prepared by using a bunch of data preprocessing and statistic toolboxes. Because raw data was incomplete, noisy and has some outliers that can cause resulting of regression models with less accuracy or fitting problems [28].

3.2.1 Data Cleaning

Missing data is a frequently encountered problem for recording datasets and can cause various problems. Mainly the statistical properties can be lost and cause a biasing effect when hyper parameters tuning [29]. Thus, it can prevent

getting the highest accuracy with tuned parameters. Besides, Python data science frameworks are not working with missing data.

Missing data handling can be grouped under three main headings [29]:

1. **Dropping Features:** This method proposes the deletion of columns. It's feasible to delete a column only if 70% of the data is missing. We applied this method for the 'Downstream (kts)' column (see Table 5) because, 100% of this variable was missing.
2. **Partial Deletion:** This method proposes deleting rows that contain any missing data. If the time series data has seasonal effects and is periodic, then particle deletion may cause a distortion in a period belonging to sampled data. But the dataset which is used in this thesis is not periodic. However, only the 'Ekonomi' variable has 35.23% missing values (see Table 5). This variable is not necessary for regression models. Therefore, the column of this variable also dropped in order to keep all the statistical properties.
3. **Imputing:** Imputation is a filling data method but has more complex structures. Mainly, imputing can be grouped into two categories as single and multiple values imputing:
 1. **Single imputing:** Imputing is done for only one value in data. The missing value can be replaced either by a mean value of a column or a regression model.
 1. **Mean value:** It can cause a bias in the model.
 2. **Regression model:** XGBoost is one of the most used regressors. The existing values are independent variables, and missing values are target variables in order to fit them. This method is less harmful to the model for the bias effect.
 - a. **Multiple imputation:** Imputing for multiple values in a dataset. The regression model in single imputation can be used iteratively. In our dataset, multiple imputation method is used via the 'IterativeImputer' method of the Python scikit-learn framework. The method basically applies a strategy using other features in a round-robin [22].

Table 5 Missing column values in percentage

Column Name	Total Missing Values	Percent Missing
Ekonomi (kg/nm)	37946	35.23
Downstream (kts)	107716	100

3.2.2 Resampling

Original data recording samples have 1 second period. But this time interval causes a long running time for our regression models and also restricts the hyperparameter tuning possibilities with more trials. Besides, downsampling can help to decrease outliers. Thus, it is substantially avoided from regression

fit problems. For these reasons, the whole raw data was downsampled into 30 seconds periods by the averaging method. In Figure 10, it can be shown that fuel consumption resampled data follows the trend of the original signal successfully. Furthermore, this resampling method substantially removes the noise and outliers of raw data.

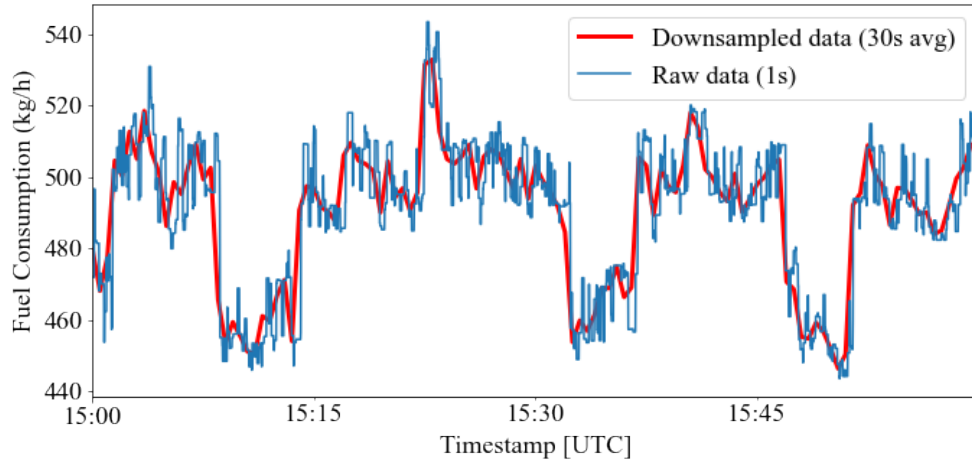


Figure 10 30 seconds down-sampling of fuel consumption raw data

3.2.3 Outlier (Anomaly) Detection

Outliers or anomalies can cause many problems in regression modeling and fitting. Outliers are directly proportional to the error of variance. The regression fitting process with many outliers can hardly decrease the RMSE value and extend the running time. Also, physical memory and processor source consumption increases drastically. Furthermore, extra bias is added to the error metrics during regression fitting.

On the other hand, in multivariable time series, directly removing an outlier also causes data losing from other variables. If the number of outliers is a significant amount, the regression analysis can no longer forecast accurately with the real world. Instead of removing, replacing with the mean value of a variable is so common in literature, however this method can easily reduce the spread of population [30].

In literature, there are many anomaly (outlier) detection methods. Briefly, the outlier detection algorithms change according to input and outlier types. In this thesis, only multivariable time series input data type is used. Outlier types can be grouped under three categories: Point, subsequent, and time series. Point outliers are generally dependent on the misbehaviors of the variable itself. Subsequent outliers are more dependent on other variables. Because multivariable time series data are composed of more than one variable, long-time lasting anomalies are generally created by a few or more variables. Furthermore, all the time series data may include some amount of anomaly [30].

In this thesis, two outlier detection techniques are used for univariate data listed below:

- Z-Score
- Inter Quartile Range (IQR)

Z-Score determines how many standard deviations will be valid in the used dataset. But it also assumes the dataset fits a Gaussian distribution. In Equation 5, Z-Score is calculated by the difference between observed and mean values divided by a standard deviation.

$$z = \frac{x - \mu}{\sigma} \quad (5)$$

In Figure 11, the field under distribution represents the amount of data in percent. The number of outliers increases by decreasing the standard deviation.

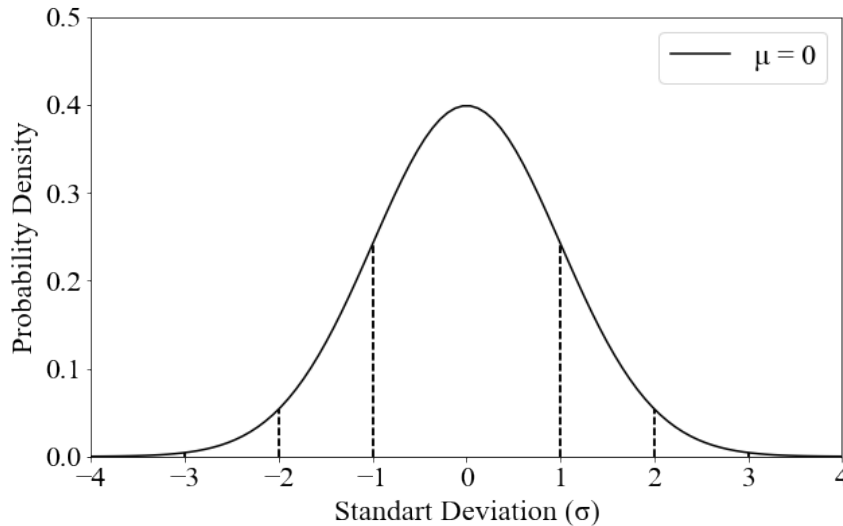


Figure 11 Standard deviations in a normal distribution

IQR methods do not use gaussian distribution. So that it can also apply to the non-parametric data. A general box plot view of IQR values can be seen in Figure 12. In this figure, Q1, median and Q3 represent the 25th, 50th and 75th percentile of the data. The smallest value in IQR is 1.5 times less than Q1. The largest value is 1.5 times larger than Q3. IQR plots are mostly used for drawing outliers in data analysis projects.

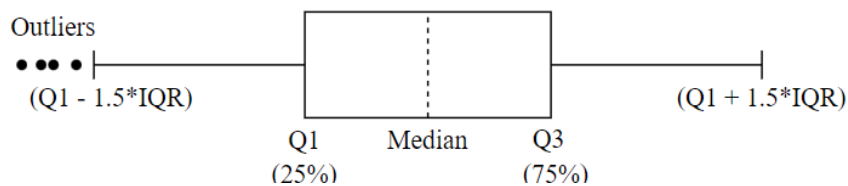


Figure 12 A general box-plot representation of IQR

The outliers of fuel consumption data can be seen in Figure 13 box plot. There seems to be a few outliers around 100 kg/h fuel consumption value. It is not possible to learn the number of outliers from a box-plot representation. Therefore, the Z-Score method is applied to the fuel consumption column by a standard deviation (σ) or threshold value 2. As a result, 20 outliers are found by the Z-Score method application (see Table 6).

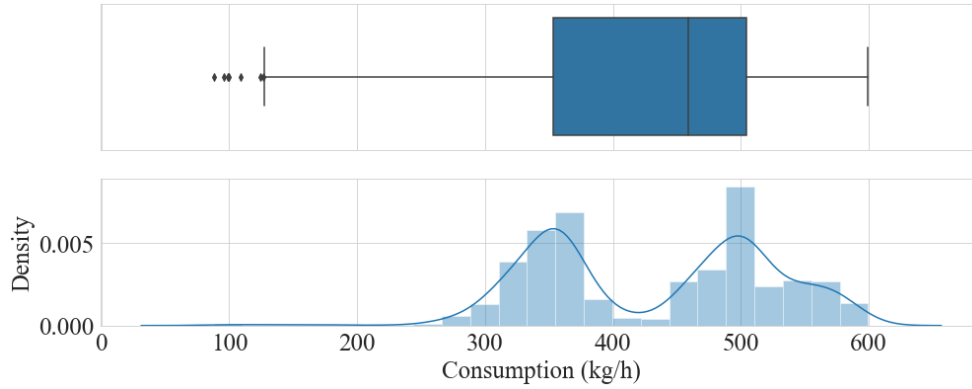
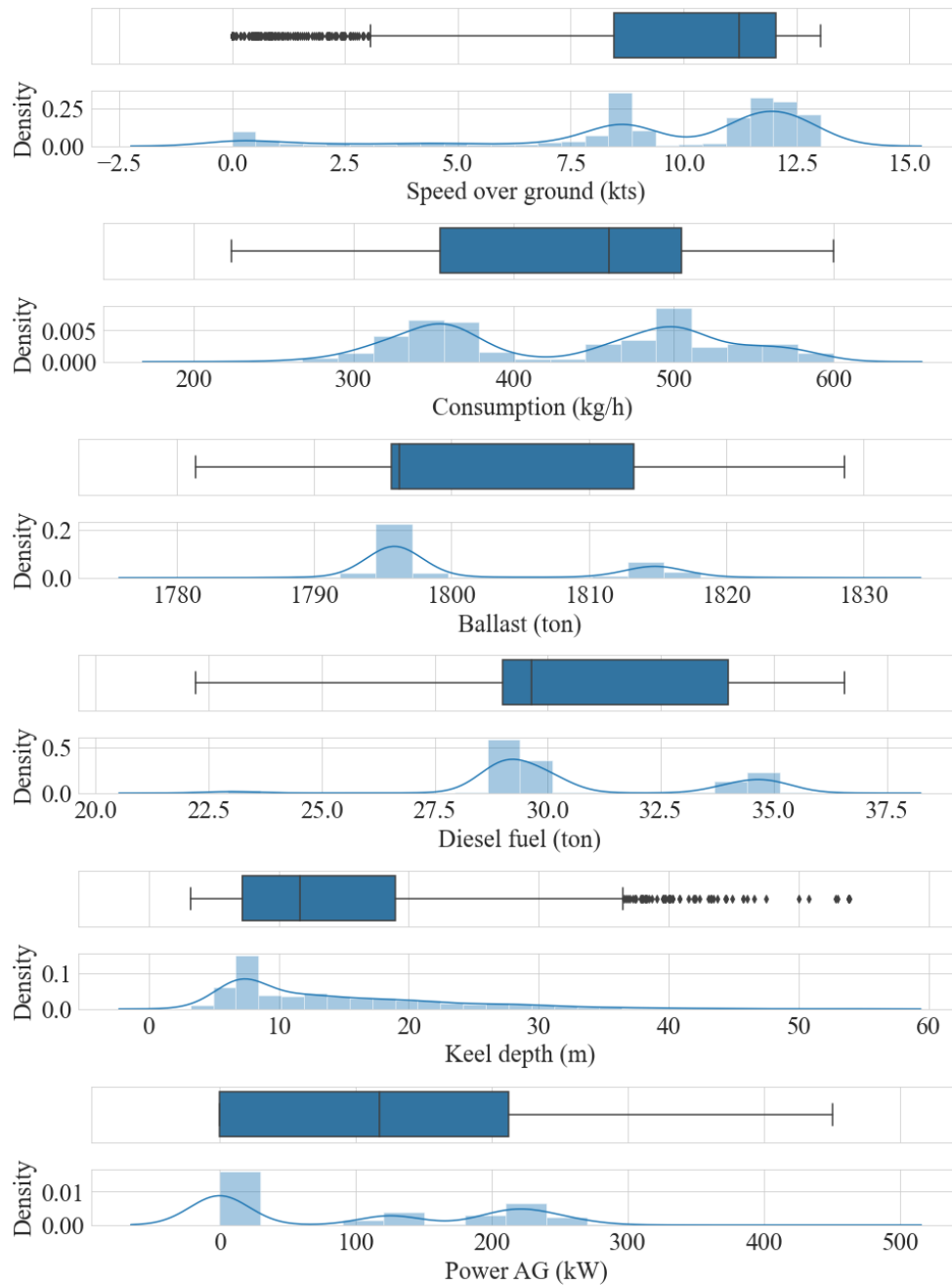


Figure 13 Fuel consumption data distribution and IQR

Table 6 Fuel consumption IQR method outliers ($\sigma = 2$)

	Timestamp [UTC]	Consumption (kg/h)
1	2021-02-10 09:49:30	124.41
2	2021-02-10 09:50:00	127.62
3	2021-02-10 09:50:30	176.86
4	2021-02-10 09:51:00	164.00
5	2021-02-10 09:51:30	148.87
6	2021-02-10 09:52:00	151.11
7	2021-02-10 09:52:30	159.08
8	2021-02-10 09:53:00	224.50
9	2021-02-10 09:53:30	223.69
10	2021-02-10 09:54:00	184.45
11	2021-02-11 05:57:00	247.81
12	2021-02-11 05:57:30	179.67
13	2021-02-11 05:58:00	127.06
14	2021-02-11 05:58:30	95.58
15	2021-02-11 05:59:00	99.36
16	2021-02-11 05:59:30	109.13
17	2021-02-11 06:00:00	98.78
18	2021-02-11 06:00:30	88.42
20	2021-02-10 09:49:30	124.41

The box-plot of the rest of all variables can be seen in Figure 14. Furthermore, it can easily be noticed that almost all of them include a large number of outliers with respect to the same standard deviation value 2. Also, most of the variables have a significant amount of skewness due to a large number of outliers. Furthermore, due to having multivariable time series, applying the Z-Score method to other variables separately will cause large data loss. In addition, decreasing data can cause losing statistical properties of real world data. In order to avoid losing data, a multivariable anomaly(outlier) detection algorithm was implemented using all the variables.



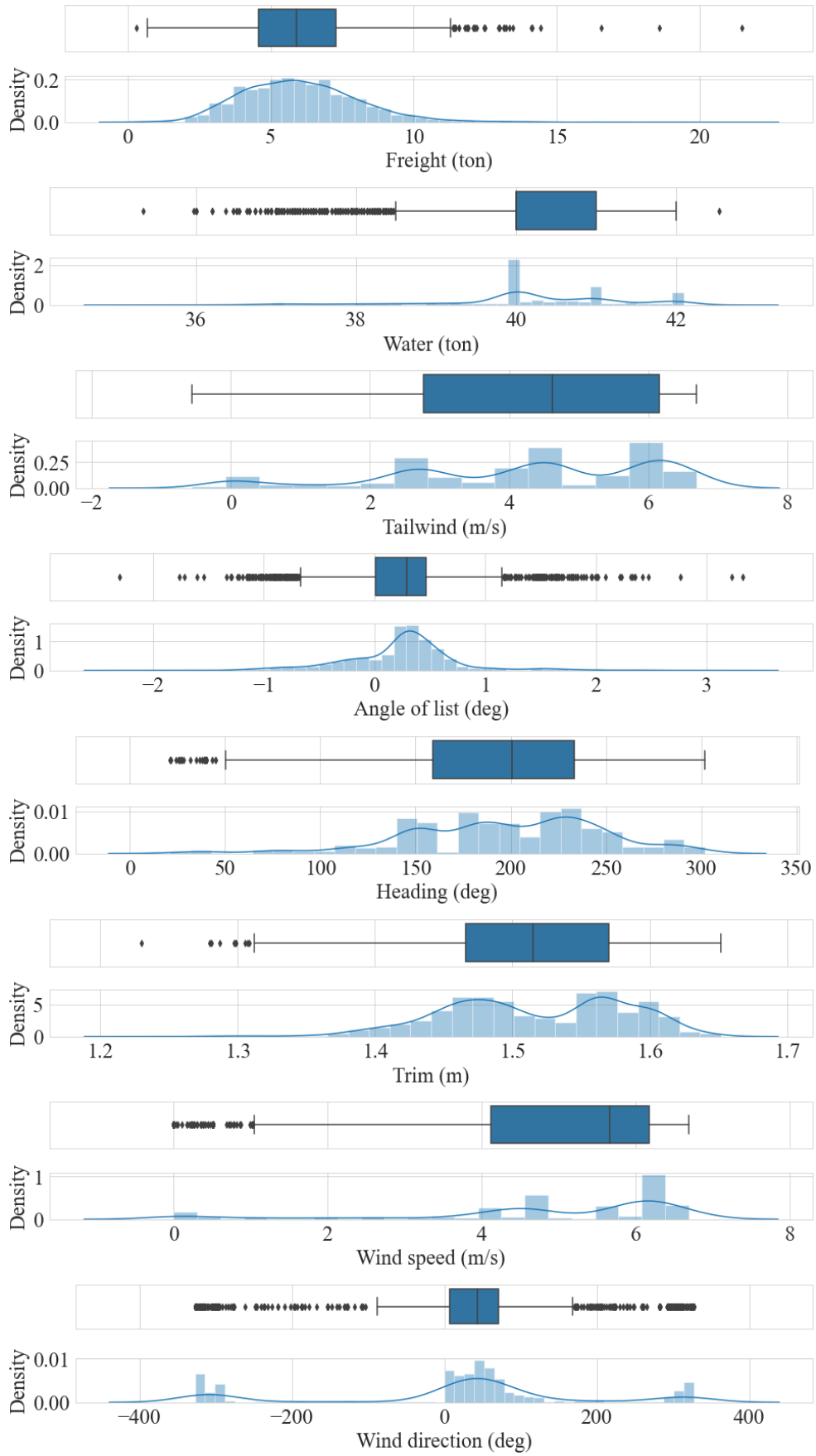


Figure 14 Histograms and IQRs of regression features

3.2.3.1 Local Outlier Factor Algorithm

The local Outlier Factor (LOF) algorithm finds the outliers by defining a degree of being outlying. Briefly, the algorithm uses a density function to compare with neighbors of the point. Because the main assumption of the LOF algorithm is that the density of outliers differs from the density of neighbors much more than non-outlier ones, on the other side, density is based on k-nearest neighbors and k-distance of them.

Figure 15 is demonstrated in order to explain the LOF algorithm. If the k-distance defines as a distance from point o to a circular line, then the reachability distances of all the objects inside this circular region are k-distance. For example, although the normal distance from the point p_1 to point o is lower than k-distance, the reachability distance of p_1 , $reach-dist_k(p_1, o)$ is equal k-distance(o) (see Figure 15). Otherwise, the reachability distance for point p_2 is directly equal to its normal distance to point o . Also, the number of all points inside this circle defines as the number of k-neighbors $N_k(p)$.

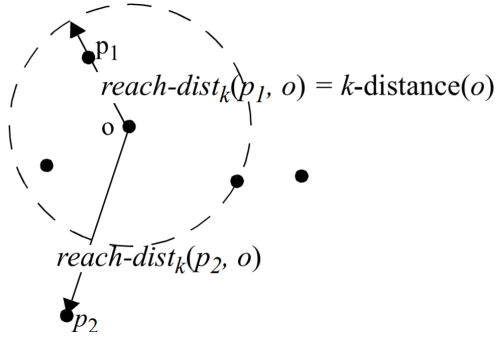


Figure 15 Reachability distances of k-neighbors [31]

Moreover, another term, local reachability density (lrd) is used in LOF to compare points and their neighbors' inverse of averaged reachability distances are formulated in Equation 6.

$$lrd_k(p) = [sum(reach-dist_k(p, o) / N_k(p))]^{-1} \quad \forall p, o \in D \quad (6)$$

Finally, the local outlier factor can be defined by using local reachability densities and the number of neighborhoods. In Equation 7, it is defined as an average of the ratio of the reachability densities of point p and the number of its neighborhood.

$$LOF_k(p) = \frac{\sum \frac{lrd_k(o)}{N_k(p)}}{N_k(p)} \quad (7)$$

In conclusion, it's obvious that if the local reachability densities of neighbors are higher than point p reachable density, then the local outlier factor will be higher for point p . It means that lower local outlier factors are closer to

assigning as an outlier. In other words, isolated points will have lower local densities than their neighbors inherently and LOF of them will be calculated lower as well.

In this thesis study, the scikit-learn framework is used to implement the LOF algorithm. All the selected ship variables listed in Table 4 are used when the proposed method is applied. The default distance metric (Minkowski) is used. Also, again the default Euclidean distance method is selected for the Minkowski metric. In order to search for nearest neighbors, an automatic method selection configuration is activated. It will automatically select one of BallTree, KDTree, or Brute Force search algorithms based on the values passed to the regression fitting method (see Appendix A for more detailed parameter settings). Besides, k-neighbors and estimated outlier proportion parameters are determined by changing them increasingly. For this purpose, 20, 50 and 100 numbers of neighbors (k-neighbors) and 1, 5 and 10 percent of estimated contamination ratios are applied respectively and iteratively together in the condition that all other variables are kept constant. Finally, the results of the LOF method are shown as scatter plots in Figure 16. The red-colored points indicate outliers. Although the 14 input variables are used as input for the LOF method, only fuel consumption and speed variables are shown in scatter plots. Because the speed variable will be used as a control parameter in the speed optimization section of this thesis. It's worth emphasizing again that the resulted outliers are determined by 14 parameters. Therefore, it's normal to be many green points that seem to be isolated extremely from neighbors but do not result as outliers exist.

At most five percent data is removed by setting contamination ratio as 5. Because the time range of source data is already limited to 1 day. Moreover, the ship variables can already show different characteristics generally in long time ranges like 1 month or more. Therefore, losing more data or statistical properties from provided raw data is avoided. Also, it can be followed in Figure 16 that increasing contamination ratio causes many outliers to bring out wrongly between normal green areas. On the other side, increasing k-neighbors results as more desired with many outliers in zero speed regions. For that reason, k-neighbors and contamination parameters are selected as 50 and 5, respectively, are shown in sub-plot (h) in Figure 16.

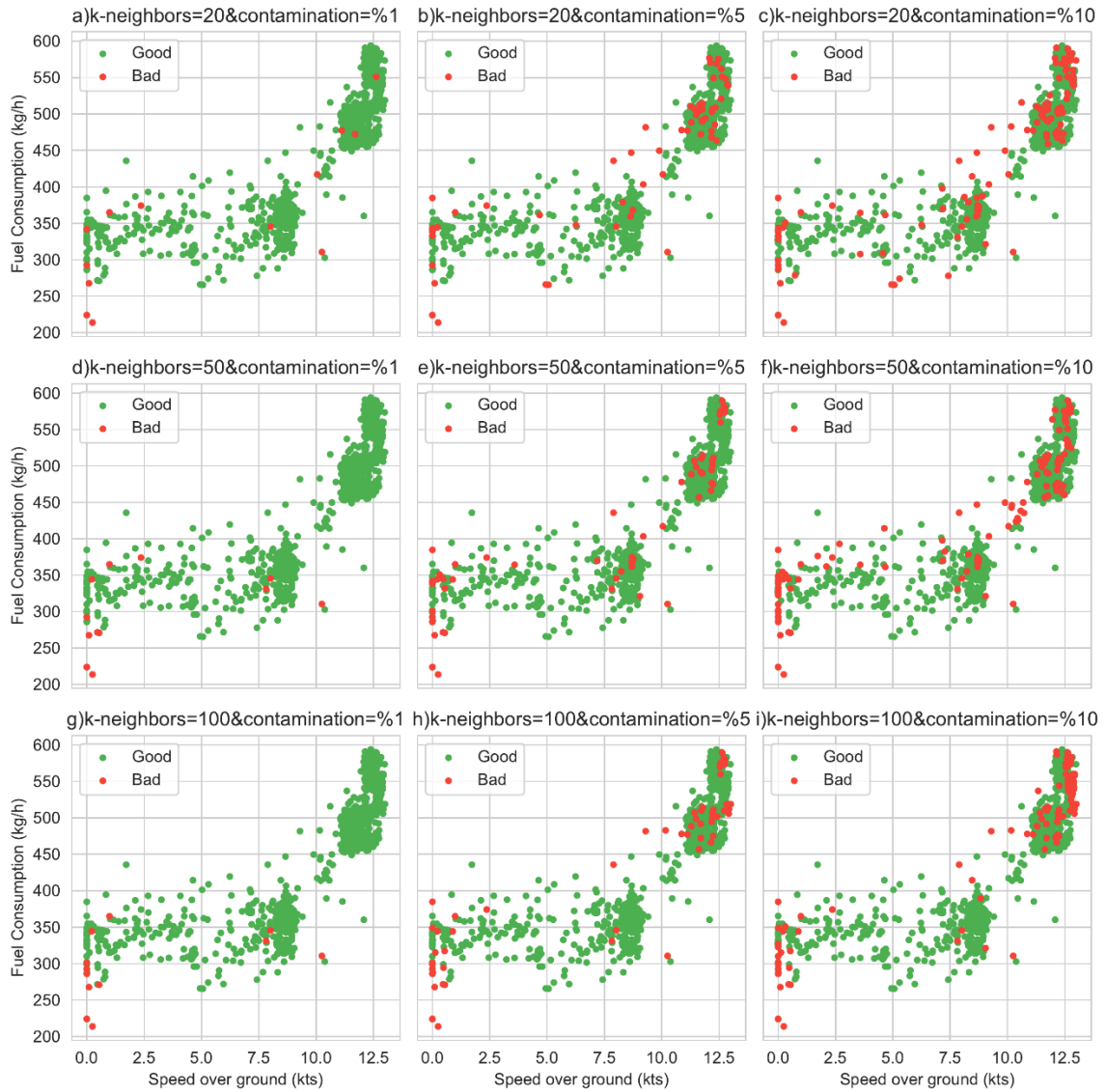


Figure 16 LOF method outlier detection results on various combinations of k-neighbors and contamination parameter values

3.3 Feature Selection

Using all the features to estimate fuel consumption can cause a lot of noise and require too much computational time and resources. However, a correlation analysis is required to determine unnecessary features [21]. By a correlation analysis, highly correlated features are considered as a single variable in the aspect of estimators [32]. Correlation coefficients (R) explain linear relationships only between two variable datasets. R-value is normalized between 1 and -1. By using the help of Figure 17, in the condition of R is equal to 1, a perfect positive linear relationship exists with X1 and X2 increase or decrease completely together. If R is equal to -1, the perfect negative linear relationship exists between X1 and X2 increase or decrease in completely opposite directions. Meanwhile, when the R is equal to 0 means that there is no linear relationship between two variable datasets [33].

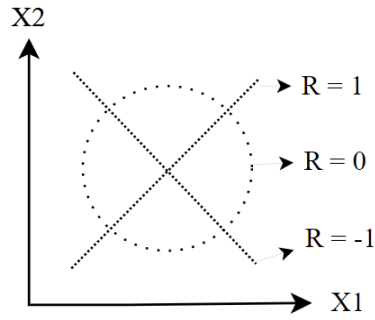


Figure 17 Extreme correlation coefficients of X1 and X2 scatter plots

Therefore, the correlation matrix shown in Figure 18 is also a square-shaped and symmetric matrix. The main diagonal elements refer to themselves and are always 1.

Speed over ground (kts)	1.00	0.78	0.34	0.46	0.55	0.17	-0.08	-0.09	0.74	0.05	-0.14	-0.77	1.00	-0.21
Consumption (kg/h)	0.78	1.00	0.51	0.58	0.65	0.32	-0.06	-0.07	0.46	-0.08	-0.36	-0.85	0.78	-0.20
Ballast (ton)	0.34	0.51	1.00	0.82	0.59	0.13	-0.30	0.47	-0.06	0.15	-0.64	-0.49	0.33	-0.18
Diesel fuel (ton)	0.46	0.58	0.82	1.00	0.48	0.07	-0.26	0.32	0.15	-0.07	-0.51	-0.61	0.45	-0.05
Keel depth (m)	0.55	0.65	0.59	0.48	1.00	0.53	-0.22	0.08	0.21	0.18	-0.38	-0.59	0.55	-0.29
Power AG (kW)	0.17	0.32	0.13	0.07	0.53	1.00	-0.24	-0.15	0.02	0.08	-0.24	-0.23	0.17	-0.19
Freight (ton)	-0.08	-0.06	-0.30	-0.26	-0.22	-0.24	1.00	-0.12	0.06	-0.09	0.31	0.06	-0.07	0.08
Water (ton)	-0.09	-0.07	0.47	0.32	0.08	-0.15	-0.12	1.00	-0.28	0.36	-0.14	0.10	-0.12	-0.08
Tailwind (m/s)	0.74	0.46	-0.06	0.15	0.21	0.02	0.06	-0.28	1.00	-0.07	0.11	-0.46	0.75	0.39
Angle of list (deg)	0.05	-0.08	0.15	-0.07	0.18	0.08	-0.09	0.36	-0.07	1.00	0.32	0.34	0.05	-0.30
Heading (deg)	-0.14	-0.36	-0.64	-0.51	-0.38	-0.24	0.31	-0.14	0.11	0.32	1.00	0.40	-0.14	0.02
Trim (m)	-0.77	-0.85	-0.49	-0.61	-0.59	-0.23	0.06	0.10	-0.46	0.34	0.40	1.00	-0.76	0.13
Wind speed (m/s)	1.00	0.78	0.33	0.45	0.55	0.17	-0.07	-0.12	0.75	0.05	-0.14	-0.76	1.00	-0.21
Wind direction (deg)	-0.21	-0.20	-0.18	-0.05	-0.29	-0.19	0.08	-0.08	0.39	-0.30	0.02	0.13	-0.21	1.00

Figure 18 Pearson correlation coefficients between features

Existing linear dependency or collinearity between regression variables can cause to estimate wrongly of their regression coefficients, giving wrong importance to regression predictions [9]. In Figure 18, its noticed at first that the correlation coefficient of ship speed over ground and wind speed are 1. So, a collinearity exists for them. But speed over ground variable will be used as a control parameter for speed optimization in this thesis. So that, instead of it, the wind speed parameter is excluded from regression analysis.

4 Modelling Evaluation and Results

4.1 Performance and Validation of Estimation Models

The performance evaluation of estimation models is done by comparing actual and predicted values. Three error metrics specified below are used in this study.

a) Root Mean Square Error (RMSE)

RMSE is one of the fundamental cost parameters. In other words, the standard deviation of the differences between predicted and observed values (refer to Equation 4). Regression models iteratively measure RMSE in each iteration and try to decrease it heuristically.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (8)$$

b) Coefficient of Determination (R^2)

R^2 or accuracy metric explains how a regression model can predict truly observed fuel consumption values. The variance comparison is done of independent and dependent variables. R^2 metric can vary between -1 and 1. Higher accuracy values are better for the model and it goes negative if the model does not follow the trend in data.

c) Running Time

Running time comparisons of regression models are also used in this study. Although all the models have different accuracy and RMSE values, all of them sufficiently give the results. The running time should be take considered when they are running on bigger datasets. In an increasing running time, the resources can be easily overloaded and cause kernel-stopping failures.

4.2 Testing the Assumptions of Linear Regression

The linearity assumptions explained in Methods section are tested in the following paragraphs.

a. Checking the linearity and additivity

In Figure 19, the scatter plots belonging to all regression variables can be seen. The Y-axis is shared for all the subplots and represents an independent variable, fuel consumption. Furthermore, the slopes of line of best fits do not have a dependency to the other independent variables [34]. So, it can be claimed by using the correlation table that speed over ground and trim

variables tend to associate more linearly with dependent variables. But, it's very difficult to claim the same thing for other variables. By referring to Figure 18, it can be concluded that the correlation coefficients of water, freight and angle of list variables are -0.04, -0.07, and -0.08, respectively. They are so close to zero that these parameters do not have any expected additivity [34] on fuel consumption prediction. In order to optimize the linear regression model backward elimination method will be used for feature elimination in the next part.

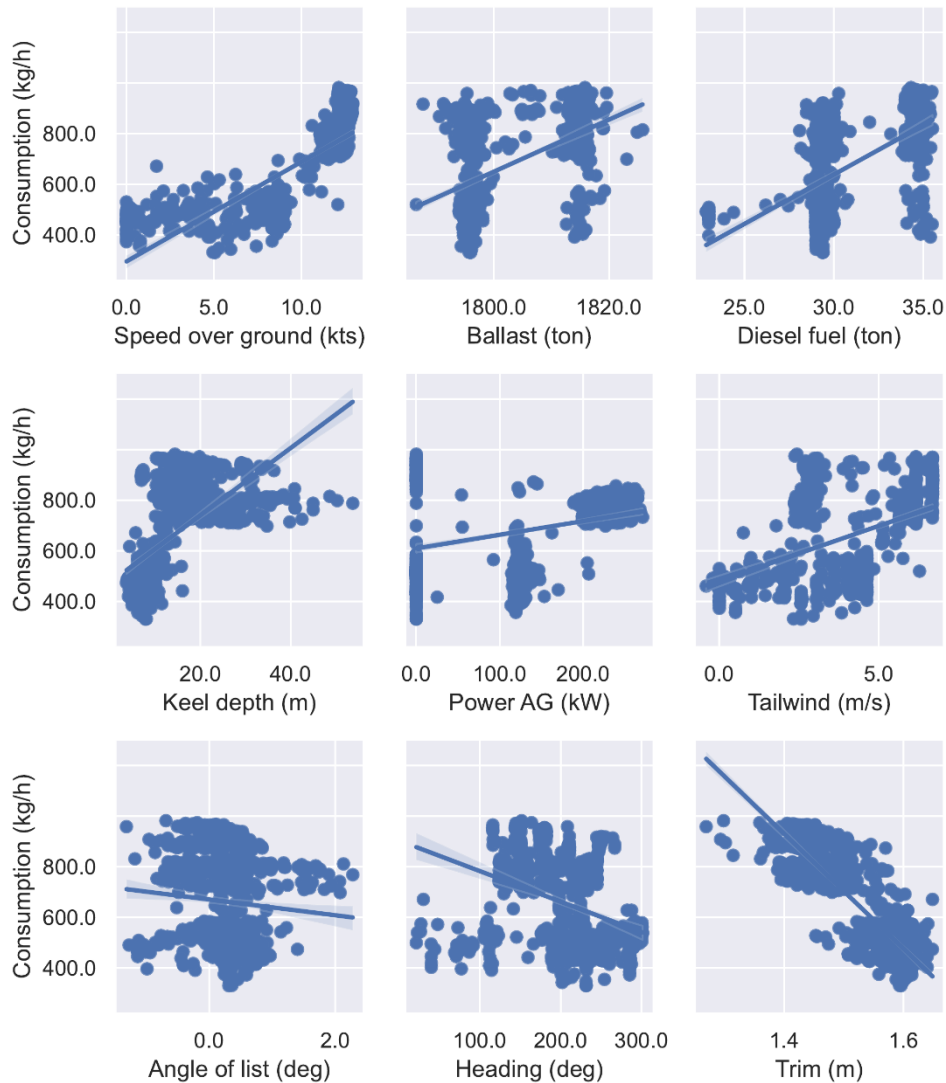


Figure 19 Scatter plots of selected features versus fuel consumption included linear regression lines of best fit

b. Checking the independence of independent variables (Multicollinearity)

We can check the multicollinearity of independent variables using bold colored cells and their correspondence variables in the correlation matrix for dependency decisions. The existence of highly correlated variables causes high RMSE errors, so the confidence interval enlarges since estimation

accuracy decreases [17]. But as a reminder, the wind speed variable was already removed from the regression analysis in the previous part. Except for that one, the R value of ballast vs. diesel and speed over ground vs. trim associations are still high. But trim is the one other important objective to investigate the effect on fuel consumption in this thesis. For this reason, they are kept in the analysis.

- c. Checking the autocorrelation, heteroscedasticity, and normality or error terms (residuals)

Suppose we continue residual analysis with the dependent variable, the linear model predictions required to bring out residuals. In Figure 20(c), the normalized residuals are not to tend in an increase along predictions axis. The distribution along zero means not in an increase/decrease trend. In other words, if the variance of the errors is increasing or decreasing over fitted values, confidence intervals will cause to get in a smaller range. This issue known as heteroscedasticity and results by giving too much importance to only a small subset of data [34]. In our case, it's not increasing or decreasing along prediction axes. Again, if the distribution errors are normally distributed, the RMSE error computation may fail to decrease it more, and confidence intervals will get too narrow or wide [34]. In our case (see Figure 20b). Finally, there is no correlation seen between residuals.

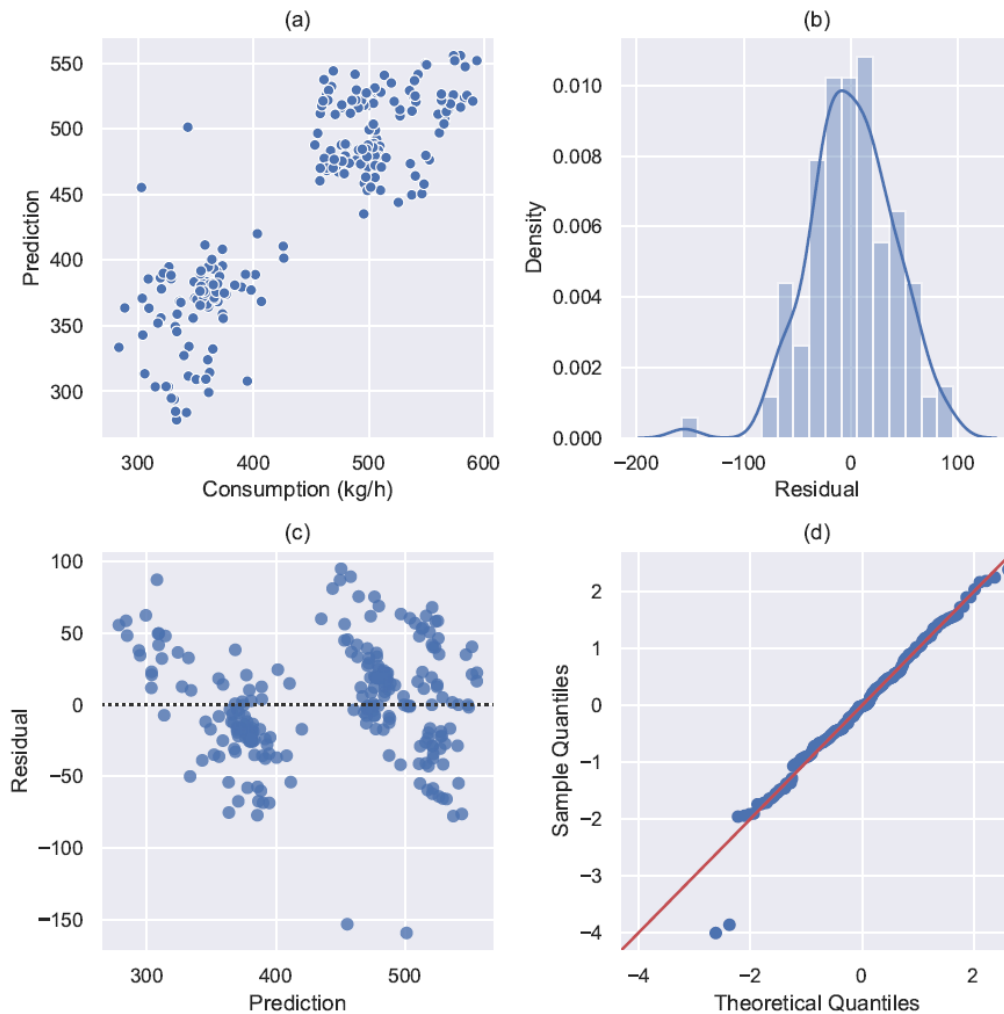


Figure 20 Prediction and residuals of linear regression (a) Prediction vs observed (b) Normal distribution of residuals (c) Normalized residuals versus predictions (d) Trend line of quantiles

Non-Linear Methods Hyperparameters

Even applied corrective transformations to the features, linear regression results do not show sufficient accuracy and RMSE values. Therefore, fuel consumption predictions using linear regression models are not reliable and feasible depending on the current dataset.

On the other hand, as seen in Figure 19, many selected features exhibit a nonlinear relationship with fuel consumption. Instead of making normal distribution assumptions on features, non-parametric methods can be used for more accurate prediction results.

Some iteration simulation results with the increasing number of neighbors are represented in Figure 21. The value of k decreases the error rate between test and training data.

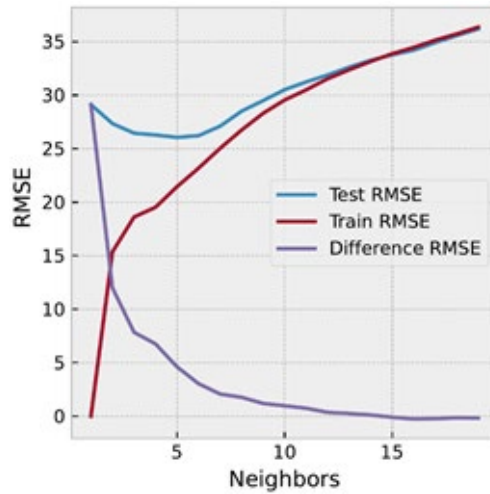


Figure 21 Test error versus neighbor numbers

By referencing Figure 22, the RF model has an overfitting problem for large depth values. The model predicts all the training data but weakly performs to predict test data. Three depth value is assigned to 8 to avoid overfitting.

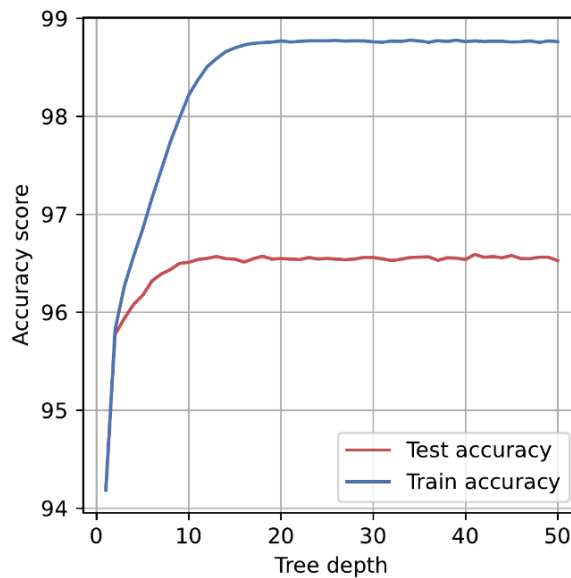


Figure 22 Three depth of tuned random forest regressor on training dataset

On the other hand, RF regression with hyperopt optimization parameters resulted in less accuracy than with RandomizedSearchCV and default RF regressor parameters (see Table 7). Hyperopt optimization accuracy evaluations for the training dataset are shown in Figure 23. Due to test and training accuracy scores of hyperopt being close to each other, the overfitting rarely occurs for hyperopt optimization.

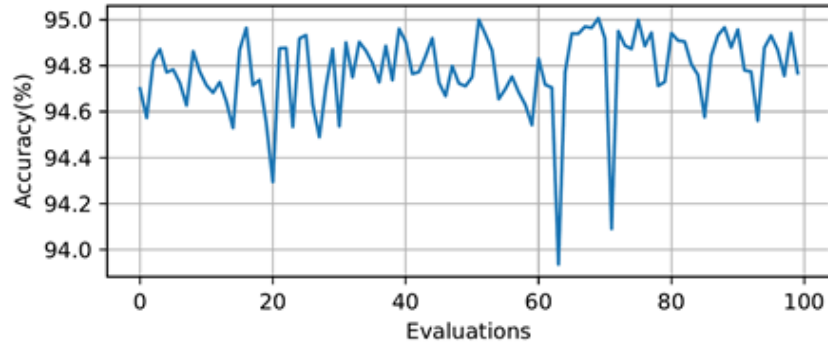


Figure 23 RF-Hyperopt trial results for training dataset

The used hyperparameters value searching ranges in this study and optimal values found by the assigned tuner are listed in Table 7.

Table 7 Model tuning hyperparameters and running times

Model	Hyperparameters Tuned	Range	Optimal Value	Train Time (min.)
LR	None			<1
SVR	None			<1
	Tuner: GridSearchCV			<120
	C	[0.1,1,10,100,1000]	1000	
	gamma	[1,0.1,0.01,0.001,0.0001]	0.1	
	kernel	['rbf','poly','sigmoid','linear']	rbf	
	degree	[1,2,3]	1	
KNN	None			<1
	Tuner: GridSearchCV			<1
	n_neighbors	[5,7,9,11,13,15,20,30,50]	5	
	weights	['uniform','distance']	distance	
	metric	['minkowski','euclidean','manhattan']	manhattan	
ADA	None			<1
	Tuner: GridSearchCV			<1
	n_estimators	[2,3,4,5,7,8,9,10,20,50,100,300]	50	
	learning_rate	[0.97,0.98,0.99,1,1.01,1.02,1.06]	1.06	
XGB	None			<1
	Tuner: RandomizedSearchCV(iteration:200)			<2
	min_child_weight	[1,3,5,7,10,15]	10	
	gamma	[0.1,0.5,1,1.5,2,5,8,15]	1	

	subsample	[0.2,0.6,0.8,1.0,1.5]	0.8
	colsample_bytree	[0.2,0.6,0.8,1.0,1.5]	1
	max_depth	[3,4,5,7,10,15]	15
	reg_alpha	[50,70,100,120,150]	50
	n_estimators	[100,180,300,1000,1500]	180
RF	None		<1
	Tuner: RandomizedSearchCV(iteration:200)		<2
	n_estimators	[5,20,50,100,300,450,500]	300
	max_features	['auto', 'sqrt']	auto
	max_depth	np.linspace(10,120,num=24)	10
	min_samples_split	[2, 4, 6, 8, 10]	2
	min_samples_leaf	[1, 2, 3, 4, 5]	2
	Tuner: HyperOpt		<15
	n_estimators	uniform(100,1000)	469
	max_depth	uniform(5,120)	16
	min_samples_leaf	uniform(1,5)	2
	min_samples_split	uniform(2,10)	4
	max_features	['auto','sqrt','log2',None]	2
GBR	None		<1
	Tuner: GridSearchCV		<5
	n_estimators	sample(200,1100,step:50)	1050
	max_features	['auto', 'sqrt']	auto
	max_depth	sample(4, 16, step:2)	12
	min_samples_split	sample(2, 20, step:1)	2
	min_samples_leaf	sample(5, 61, step:5)	5
	Tuner: RandomizedSearchCV(iteration:200)		<5
	n_estimators	rand(low:100, high:1200)	460
	max_features	rand(low:5, high:20)	8
	max_depth	rand(low:2, high:30)	9
	min_samples_split	rand(low:2, high:100)	55
	min_samples_leaf	rand(low:2, high:25)	22
	learning_rate	rand(low:0, high:1)	0.0425
	subsample	rand(low:0, high:1)	0.823

4.3 Model Performance Evaluation

The evaluation process is carried out with the corresponding training, and test dataset performance metrics resulted in Table 8 and Table 9, respectively.

As a result, the R^2 values of the linear regression model are around 80 percent, so the model weakly explains the fuel consumption compared with the studies in the literature [13, 21, 35]. Additionally, in Figure 24, the validation and training accuracy scores of LR are converge to a value that is quite low with increasing size of the training data. Thus, it seems not to benefit much from adding more training data.

Even applied corrective transformations to the features, linear regression results do not show sufficient accuracy and RMSE values. Therefore, fuel consumption predictions using linear regression models are not reliable and feasible depending on the current dataset.

Also, the R^2 score of the SVR model is worse than the LR model. Also, RMSE values are less than the linear regression model (see Table 8) and R^2 scores are still weakly explains the fuel consumption by comparison to the studies in the literature [13, 36]. On the other hand, the performance metrics of KNN resulted better than LR and SVR models. Also, the KNN model in this study explains the fuel consumption better compared to the literature [13]. In addition, SVR and KNN models increases accuracy values by adding new training data (see Figure 25), and the validation and accuracy scores close to each other through adding new data. So, they are fitting good with the data. On the other hand, after hyperparameter tuning of KNN and SVR, the accuracy and validation scores of learning curves are still high but have some gap between them. So, some overfitting was observed for them. Especially for less training data, the training accuracy score of the SVM is much greater than the validation score. So, adding new training data will seems to increase generalization.

The accuracies of VR and SR ensemble models are close to each other for training and testing datasets. VR training accuracy (94,55%) is slightly higher than SR (93,53%) for test dataset. But VR and SR ensemble models still have a gap between training and testing scores and thus have an overfitting problem.

Eventually, RF has improved the training accuracy value from 98,07% to 98,39% after tuning the hyperparameters with a random search optimizer (see Table 7). Also, testing accuracy has improved from 94,48% to 94,63% and it seems to be no overfitting occurred for RF (see Table 9). So, RF is selected for speed optimization evaluation in the next chapter.

Table 8 Performance result of models for the training dataset

Model	RMSE	R ² Score (%)
SVR	41.37	77.23
LR	38.41	80.38
KNN	28.31	89.34
Tuned ADA (Grid Search)	18.53	95.43
Tuned SVR (Grid Search)	15.11	96.96
ADABoost	15.02	97.00
GBR	13.54	97.56
RF	12.04	98.07
SR	11.06	98.37
Tuned RF (Random Search)	11.00	98.39
XGBR	9.85	98.71
VR	7.87	99.18
Tuned RF (Bayesian)	7.84	99.18
Tuned GBR (Grid Search)	5.09	99.66
Tuned GBR (Random Search)	2.01	99.95
Tuned XGB (Random Search)	1.70	99.96
Tuned KNN (Grid Search)	0.01	99.99

Table 9 Performance results of models for test dataset

Model	RMSE	R ² Score (%)
SVR	41.81	75.33
LR	39.63	77.83
KNN	29.50	87.72
Tuned KNN (Grid Search)	25.57	90.78
Tuned SVR (Grid Search)	24.86	91.28
Tuned ADA (Grid Search)	23.44	92.25
SR	21.41	93.53
Tuned RF (Bayesian)	21.07	93.73
Tuned XGB (Random Search)	20.78	93.90
Tuned GBR (Random Search)	20.75	93.93
ADABoost	20.62	94.00

GBR	20.44	94.10
RF	19.77	94.48
VR	19.66	94.55
XGBR	19.61	94.57
Tuned GBR (Grid Search)	19.58	94.59
Tuned RF (Random Search)	19.52	94.63

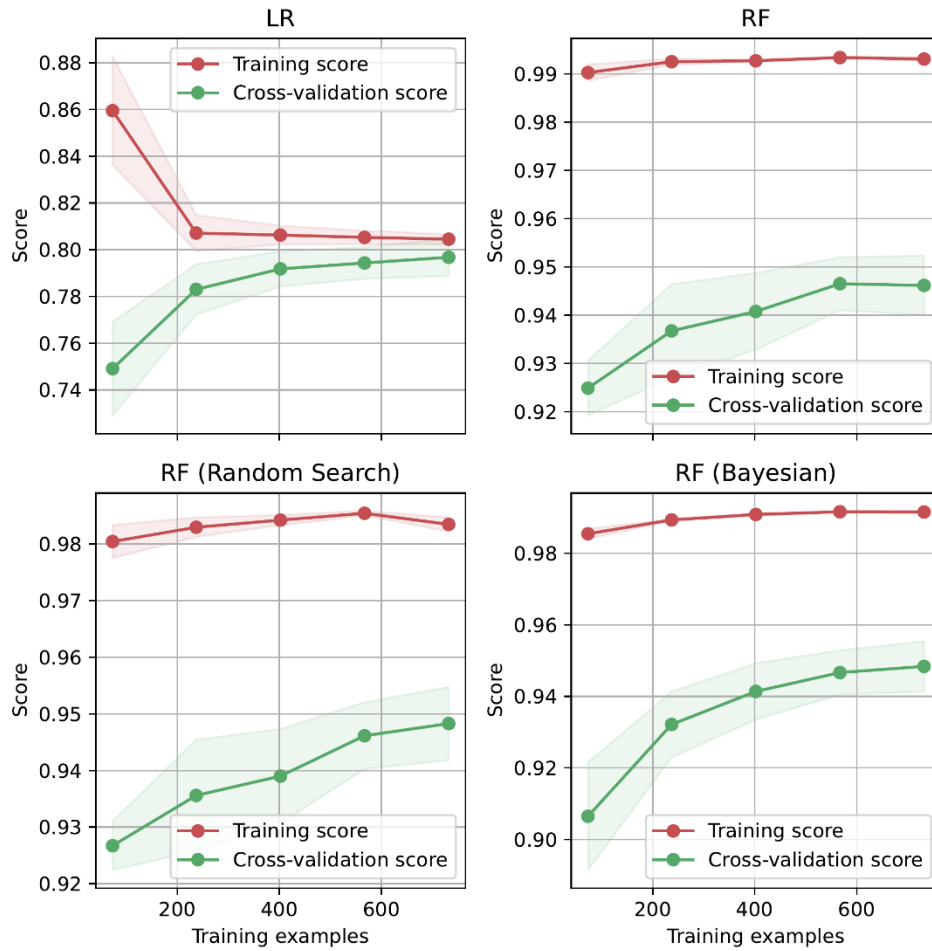


Figure 24 Learning curves of LR and RF models

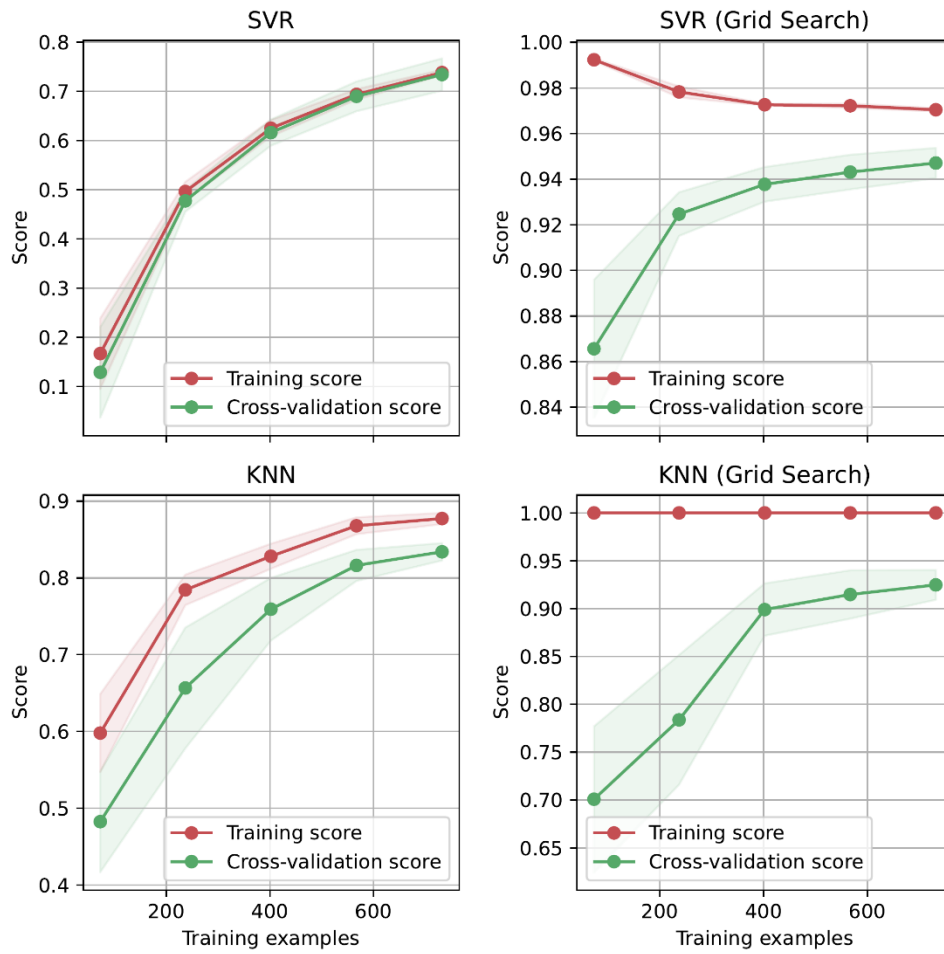


Figure 25 Learning curves of SVR and KNN models

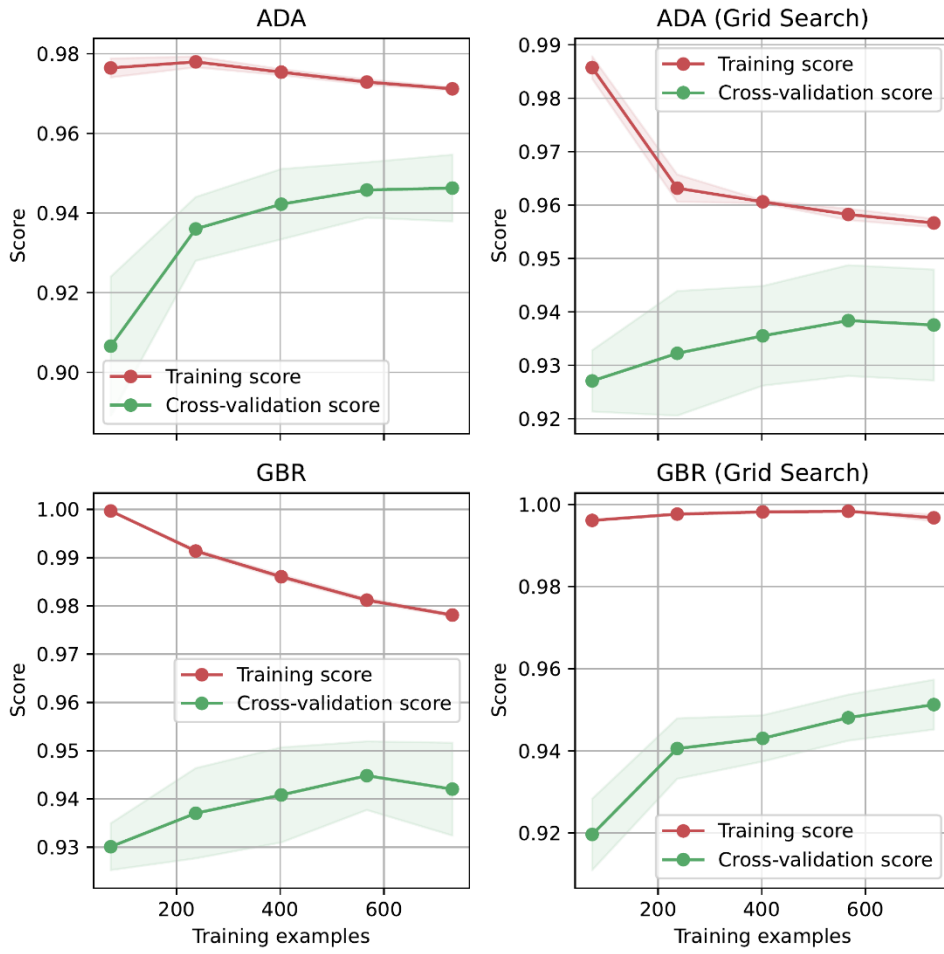


Figure 26 Learning curves of ADABOOST and GBR models

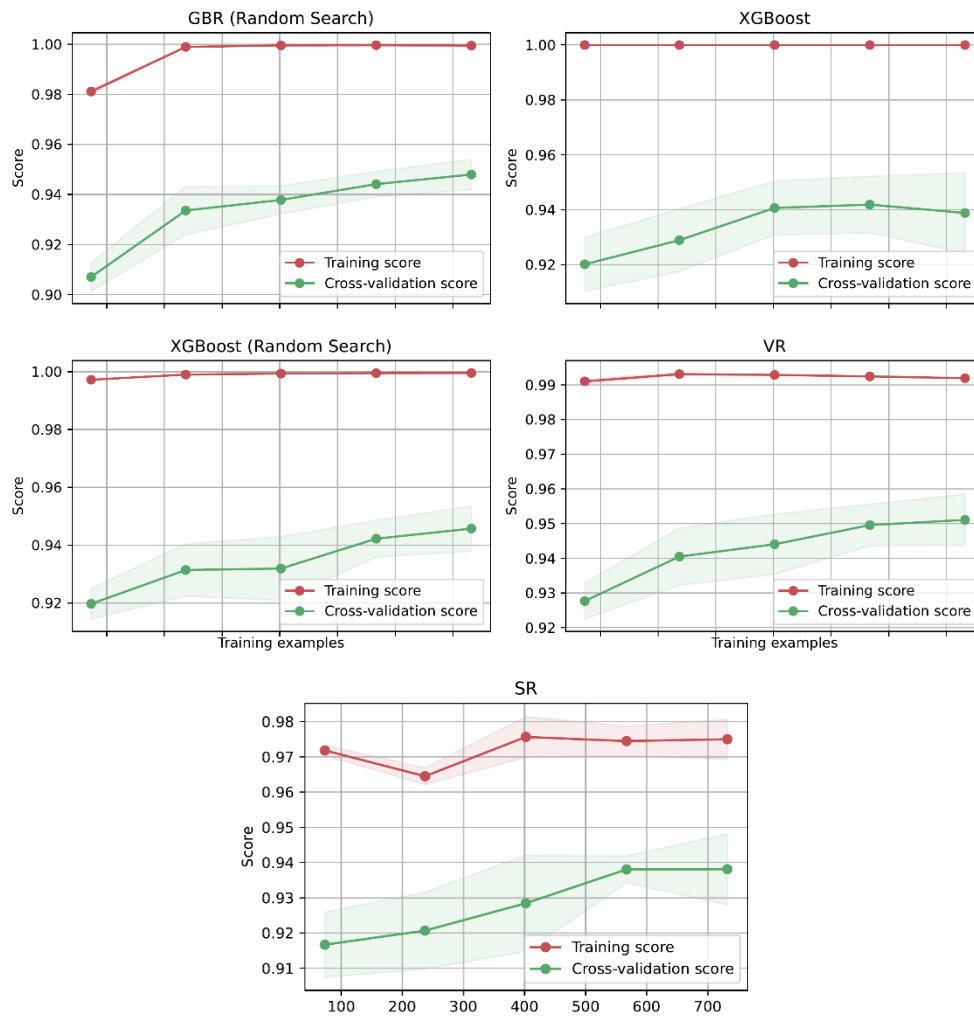


Figure 27 Learning curves of XGBoost, VR and SR models

Predicted versus measured fuel consumption (kg/h) of the best accurate models on the test dataset are presented in Figure 28.

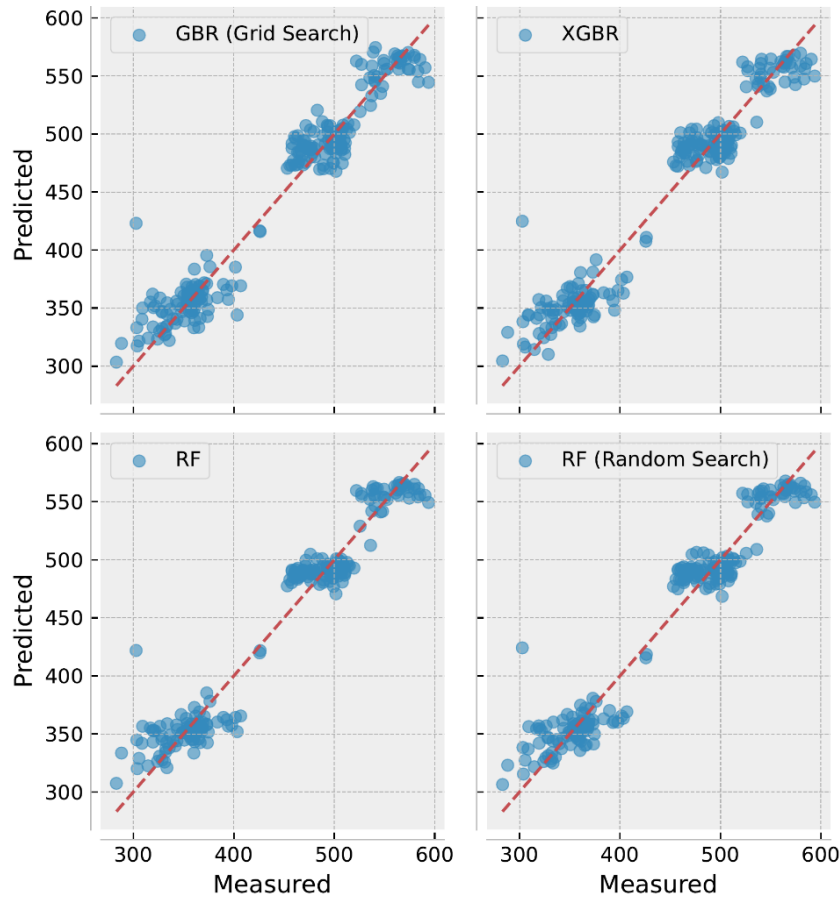


Figure 28 Predicted versus measured fuel consumption (kg/h) of tested top 4 models

4.4 Parameter Tuning of Genetic Algorithm

The performance of genetic algorithms highly depends on parameter settings [37]. All the tuning parameters used in GA are listed in Table 10. Speed decrease is initialized with a random number between 2 and 6 knots and speed decrease time is also initialized with another random number between 30 and 300 minutes for a random approach. On the contrary, they are initialized manually with constant numbers for the manual approach. So, the random approach has had more genetic variety in comparison to the manual approach.

Table 10 Tuning parameters in manual and random initializations

Parameter	Manual Initialization	Random Initialization
Mutation probability	Yes	Yes
Crossover probability	Yes	Yes
Standard dev. of mutation	Yes	Yes
Speed decrease time	Yes	No
Speed decrease	Yes	No
Individual number	Yes	Yes

Top speed	Yes	Yes
Remaining speed	Yes	Yes

Using different setting values for parameters causes a different amount of fuel consumption reduction. Another optimization study is required for the best combination of all parameters. Instead, in this study, numeric and visual comparisons of parameters have been preferred.

Besides, in this study, random and manual approaches are used for speed profile modification, so, two different tuning of parameters study are implemented to show a comparison between random and manual initialization methods of individuals.

4.4.1 Random Initialization

In the random initialization approach, firstly, the speed profile modification parameters (see Figure 3) are randomly selected to investigate better global optimal points. Speed decrease is selected randomly a value between 1 and 6 knots, and speed decrease time is between 30 and 300 minutes. Thus, a high level of genetic diversity is produced before the genetic algorithm is executed.

Afterwards, the different numbers of mutation and crossover probabilities, standard deviation, top speeds and individual numbers are used as control parameters while others are constant. Finally, the fuel consumption results are compared visually.

GA parameter tuning is applied manually in four steps. In Table 11, all the control and constant parameters are listed for every step. In the first step, mutation and crossover probability control parameters are changed and the GA runs repeatedly for 7 different combinations of them in between 10% and 90% of probabilities. In the second step, the best parameters of mutation and crossover probabilities are used and mutation standard deviation (sigma) control parameter is changed in four times from 0.2 to 1.5. In the third step, the individual numbers are changed three times as 150, 300 and 500. Finally, the speed limit and remaining speed parameters have changed together in six different combinations.

Table 11 Control parameters and constants in randomly initialized GA

Steps	Mutation probability	Crossover probability	Sigma	Population size	Speed limit (mile/h)	Remaining speed (mile/h)
1	variable	variable	1.5	300	13	13
2	20%	90%	variable	300	13	13
3	20%	90%	1	variable	13	13
4	20%	90%	1	500	variable	variable

4.4.1.1 Parameter Tuning Results

In this study, GA was run repeatedly after changing control parameters. The objective function of minimizing or fitness values of GA simulations is fuel consumption variable. The parameter tuning simulations have been completed in four steps and the results are presented in Figure 29. The parameter results are discussed in the following paragraphs.

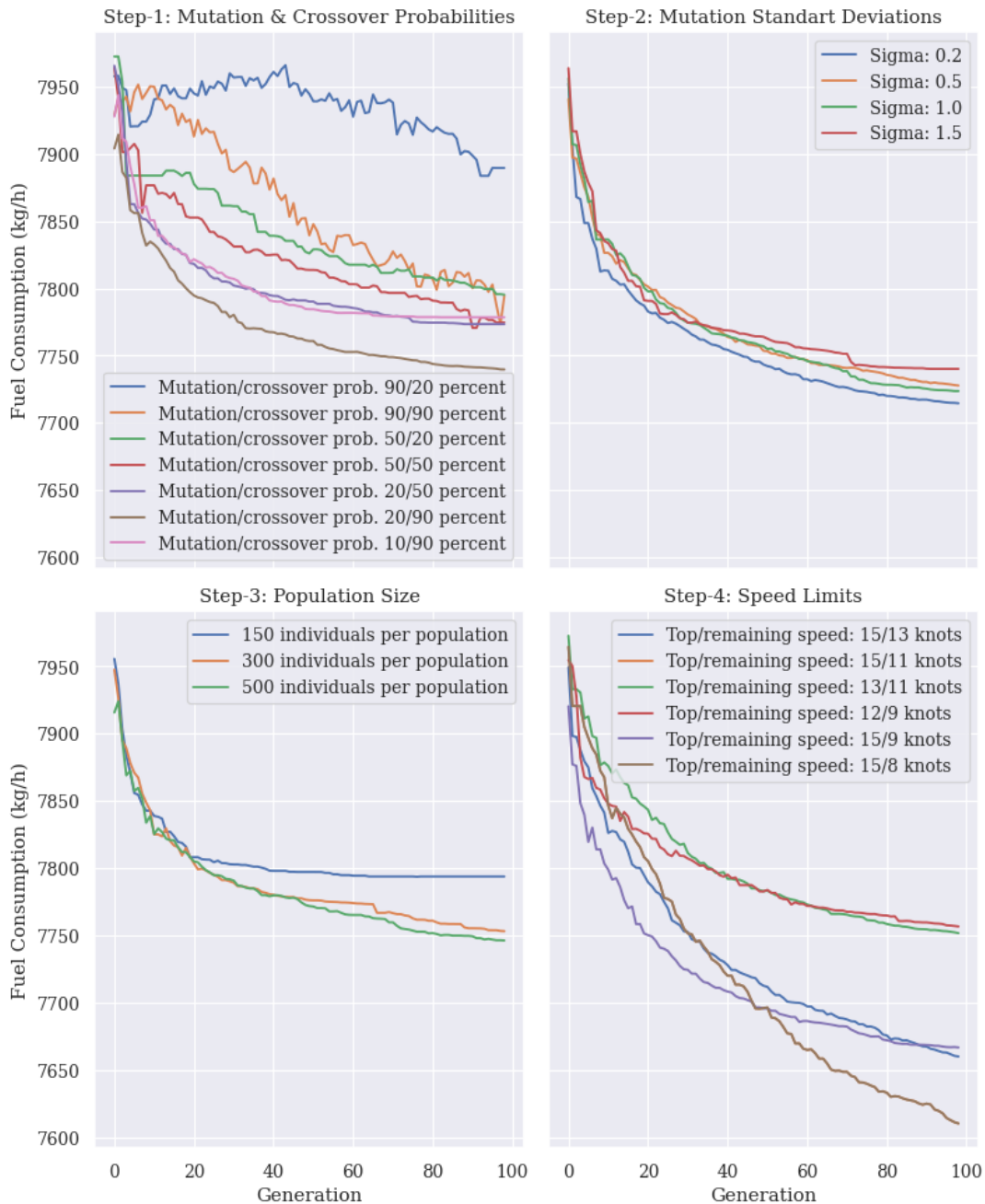


Figure 29 Parameter tuning results of random approach

Mutation and crossover probability: In Figure 29, low mutation rates are observed to show better performance together with high crossover rates. Anyway, too low mutation rates have caused more fuel consumption. In addition, by referencing Figure 30b, it is observed that increasing the

mutation rate causes unstable speed profiles. On the contrary, a higher crossover rate keeps the speed profile stable during search for global minimal point. Furthermore, the crossover operator looks all over the speed profile with equal probability (see Figure 30a).

So, 20% mutation and 90% crossover rates are chosen in a random approach for the next tunings.

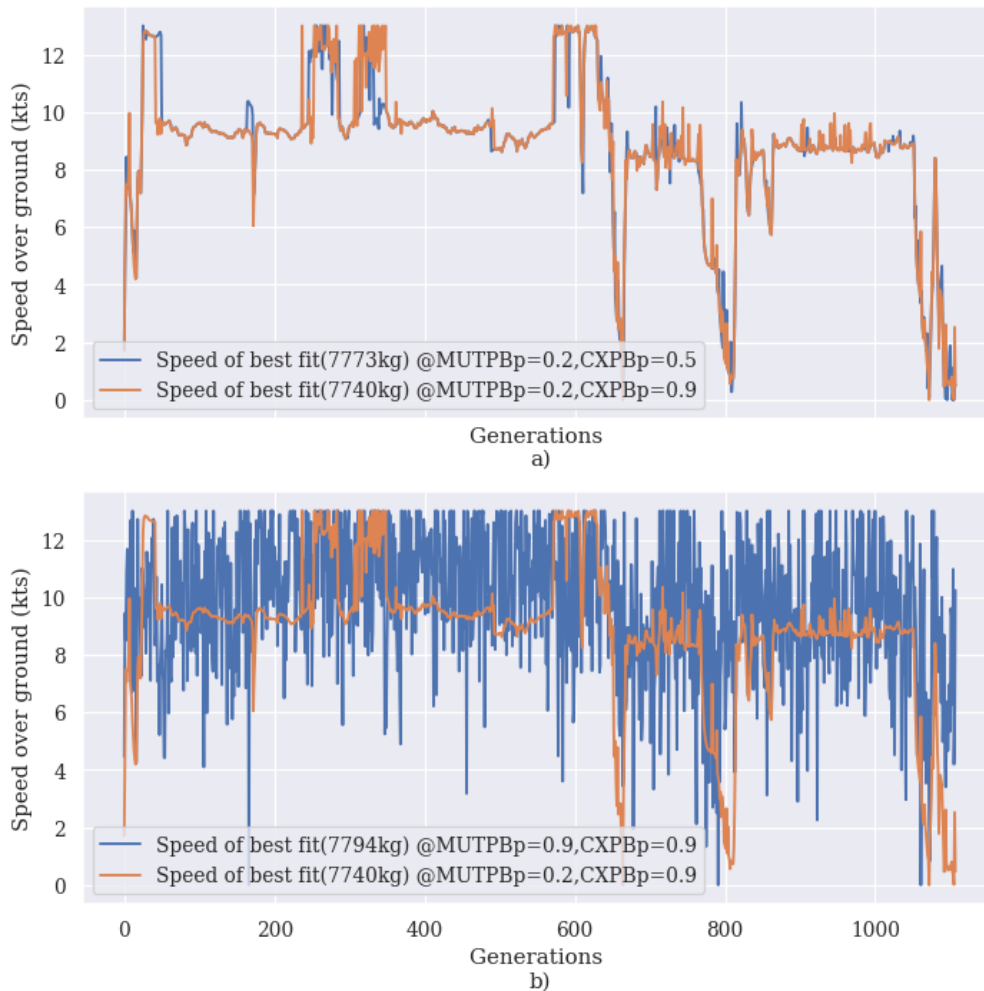


Figure 30 Speed profiles of best fits in random approach step-1, a) crossovers and b) mutations

Mutation standard deviation (sigma): Although it doesn't affect the results significantly, increasing the spread of ship speed (inverse of sigma) resulted in higher performances. The sigma parameter does not have the same effect every time. Some cases shown in Figure 29 resulted in the opposite direction. So, for the next tunings, the sigma parameter has been assigned to 1.

Besides, in Figure 31, the speed profiles of best-fitted individuals of populations with different sigma values and actual voyage speed profiles are represented together. It is observed that increasing the sigma values makes the genetic algorithm more eager to look forward to more fuel savings by changing speed values more drastically. Furthermore, as seen in the gray line

window frame in Figure 31, with increasing the sigma values, the genetic algorithm gives up seeking a solution for fuel consumption saving via small changes in speed. Thus, the optimization study has overcome being stacked in a local minimum point and continues to search for the global minimum point.

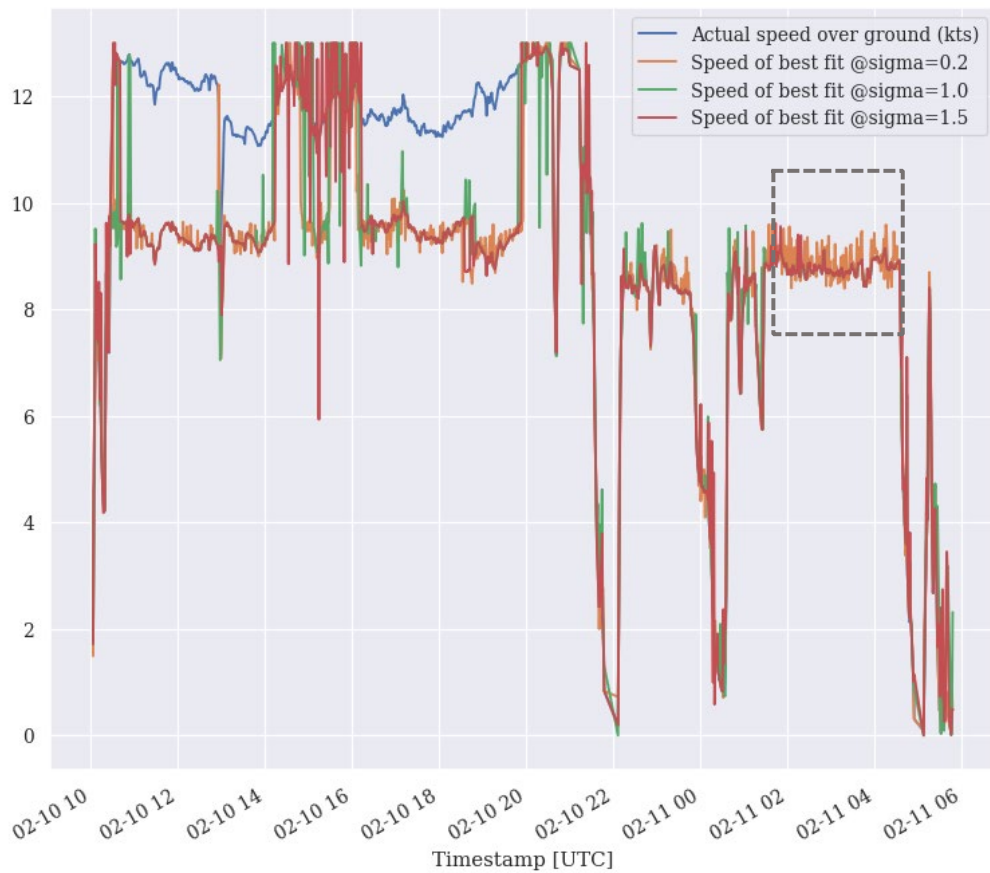


Figure 31 Speed profiles of best fits in random approach step-2, mutation standard deviations

Besides, the speed signals of the best individuals have a lot of noise for practical usage. So, a low pass filter is required to eliminate the high frequency signals in them. Therefore, an exponentially weighted moving average mean filter in Python Pandas library is applied with a very low smoothing factor (α) as 0.1 for the next figures.

Population size: By referring to Figure 29, increasing the number of individuals in a population always shows better GA performance in this study and 500 individuals showed the most efficient results.

Top/remaining speed: By referencing Figure 29, it is observed that increasing the speed limit of the voyage while lower remaining speeds are selected, result more fuel savings. Also, increasing both of speed limit and remaining speed also results in less fuel consumption than decreasing the speed limit.

Besides, it is required to define a speed for the remaining distance in GA. Because making random modifications to the speed profile can extend or shorten the distance, but longer distances are eliminated in GA. So, the remaining distances can be zero or more at the end of the evaluation step of GA.

The speed of these additional distances and the top speed limit both affect each other. Reducing the top speed limit while the remaining speed is fixed causes fewer fuel savings (see Figure 32a). On the other hand, the same behavior is not valid or vice versa. For example, while the top speed limit is fixed to 15 knots, 15 or 9 knots remaining speed cause more fuel consumption than 11 knots (see Figure 32b). In brief, the top speed limit is nonlinearly correlated with remaining voyage speed in GA and another optimization routine is required to find the best combination.

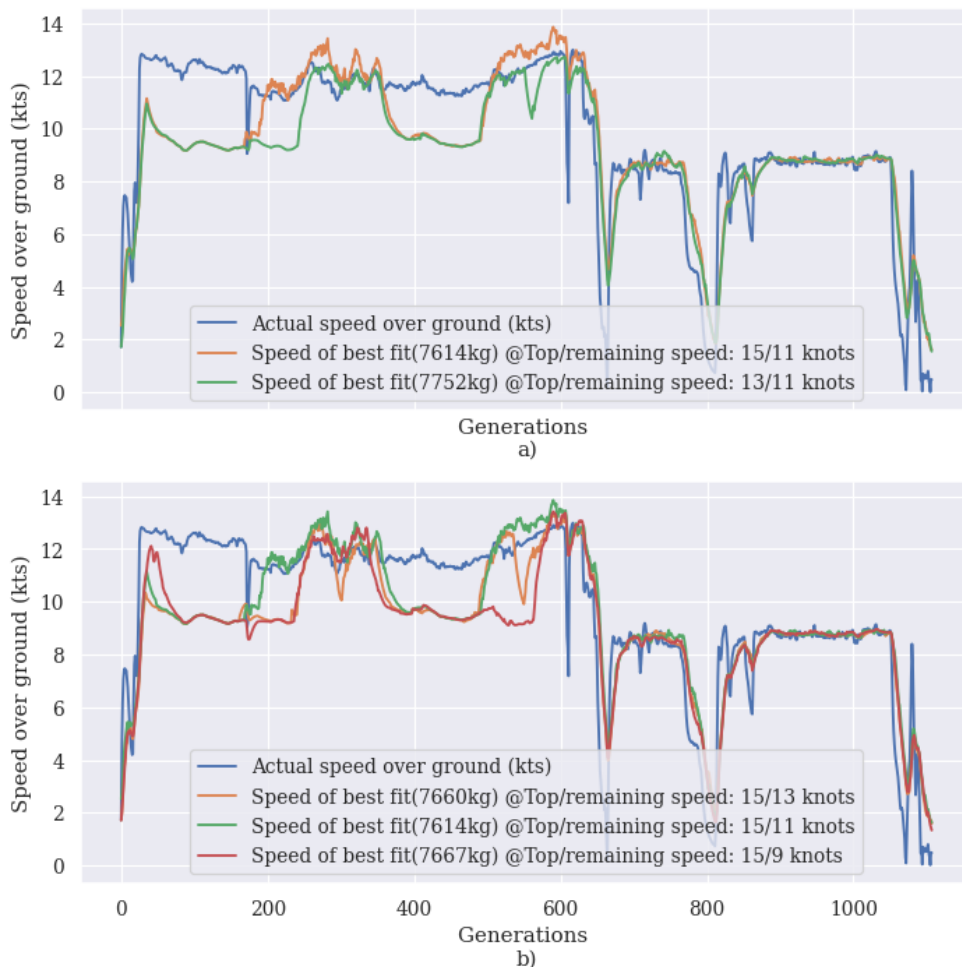


Figure 32 Speed profiles of best fits in random approach step-4, changing speed limit (a) and voyage speed for remaining distance (b)

ETA Delaying Cases

Many studies in maritime literature use a certain time window for voyages and investigate optimum speed only for trips without any delay of arrivals. But in real life, the delays are very common and voyage speeds are affected

by delay penalties. Moreover, except for weather conditions, extra detouring due to inconsistencies of bunkering costs, waiting costs due to port congestion and etc., problems causing cost inefficiency for ship operators [15]. However, hitting the two targets with one arrow is possible if a delay in ETA is used. Time delaying for port arrival time can help to avoid these problems and also save a significant amount of fuel consumption.

For these purposes, in this study, one and two hours delays of ETA are applied in the optimization routine. The GA simulated 1 and 2 hours delays under 15 knots fixed top speed limit conditions. But, different amounts of remaining speed values are taken into account to show their effect on fuel consumption. Decreasing remaining speed had always increased fuel savings at the end of GA generations completed for 1 hour delay conditions of ETA (see Figure 33a). But it is not valid for 2 hours delaying conditions. The minimum fuel consumption is occurred not for the lowest remaining speed condition. Anyway, GA implementation successfully searches minimum fuel consumption point by decreasing the voyage speed for the remaining distance at the same time. So that the remaining speed, top speed limit and delaying of ETA need to consider in another optimization algorithm to find the best combination of them.

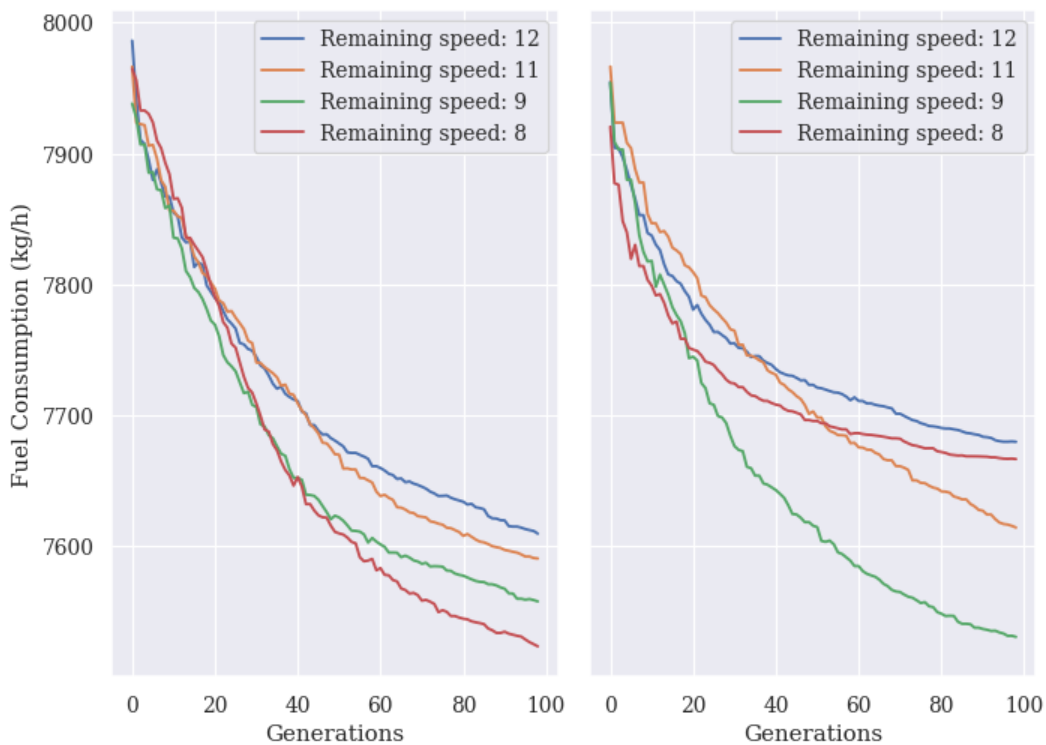


Figure 33 Fuel consumption results for 1 hour (a) and 2 hours (b) ETA delays

4.4.2 Manual Initialization

In this section, all the other parameters used in the random approach are implemented the same way except for the initialization of individuals. The speed decrease and its time interval are changed manually to create

individuals in a population. So, the new control parameters are added to others in random approaches and they are presented together in Table 12. By the way, the starting position of modification in speed profiles is always chosen randomly in both random and manual approaches.

Table 12 Control parameters and constant in manually initialized GA

Steps	Mutation probability	Crossover probability	Sigma	Speed decrease time (mins)	Speed decrease (kts)	Population size	Speed limit (mile/h)
1	variable	variable	1.5	30	2	300	13
2	20%	90%	variable	30	2	300	13
3	20%	90%	1.5	variable	2	300	13
4	20%	90%	1.5	15	variable	300	13
5	20%	90%	1.5	15	3	variable	13
6	20%	90%	1.5	15	3	500	variable

4.4.2.1 Parameter Tuning and Delaying Results

The GA simulations are run multiple times by changing the control parameter of belonging steps shown in Table 12. Also, the fuel consumption results of simulations are represented visually in Figure 34. So, by referencing visual results, the evaluations of manual initialization results are given step by step in the following paragraphs.

Briefly, parameter tuning of mutation and population size results has shown completely same results with random initialization with 20/90 percent of mutation/crossover probabilities and 500 individuals of a population resulted in more fuel savings (see Figure 34a/e).

Mutation standard deviation (sigma) values show better results when the direction of increasing and so sigma value is selected as 1.5 for the next tuning steps (see Figure 34b).

Speed reduction and its time interval are gradually increased from 15 mins to 120 mins and 1 knot to 4 knots, respectively. Thus, the lower speed reduction times provided better results with higher speed reduction amounts (see Figure 34c/d).

As a significantly important finding, the remaining distance can increase only if the speed reduction amount and its time interval increase in the manual initialization method. Of course, the time delay for the remaining distance also depends on the speed to be chosen. Finally, the speed limits are applied as GA constraints. Meanwhile, two hours delay for speed limits is used and higher speed limits resulted in more fuel savings in this study (see Figure 34f).

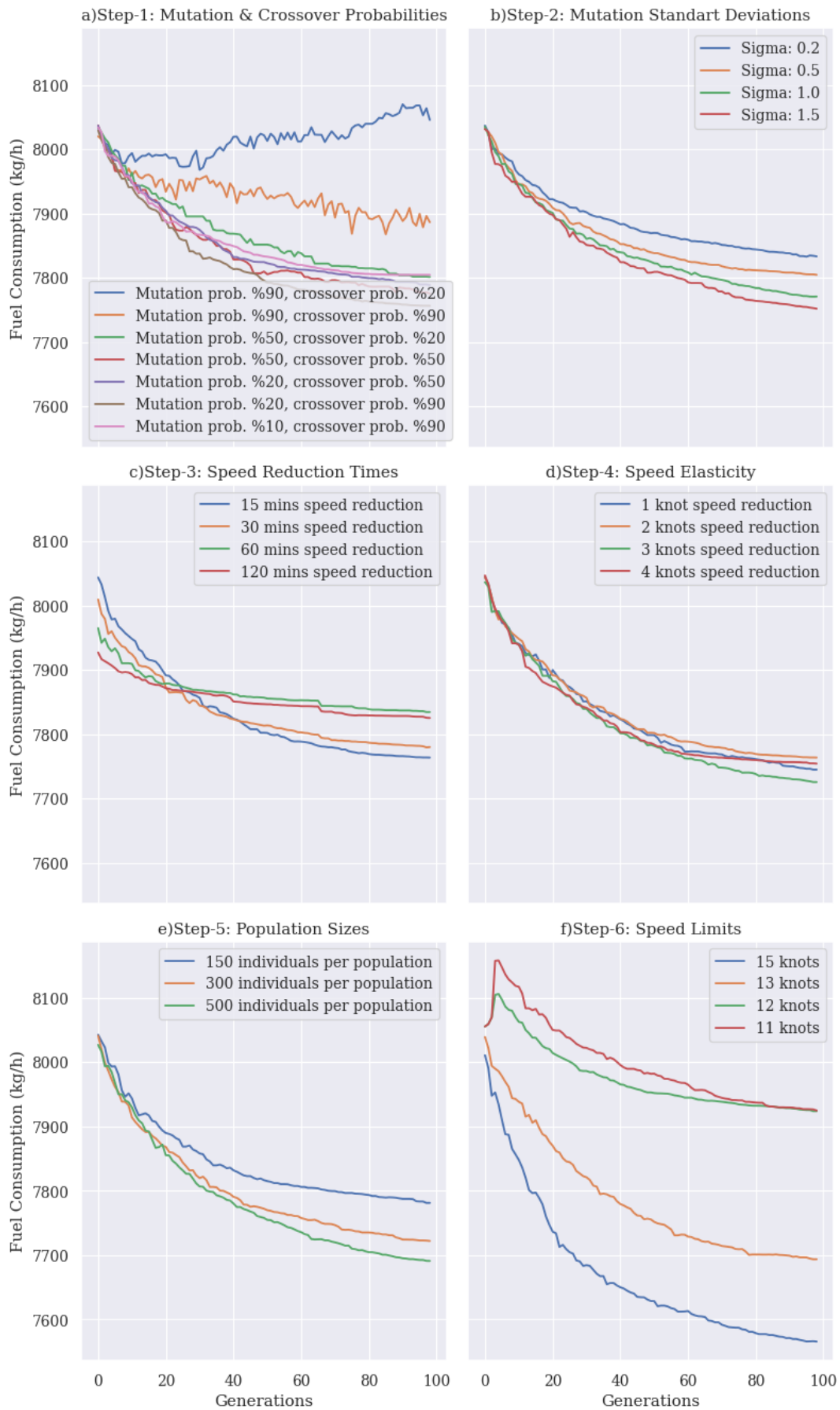


Figure 34 Parameter tuning results of manual approach

It is significantly important that the manual initialization parameters of speed decrease and its time interval should be updated according to desired delay time. Because GA performance highly depends on individual initialization [16] and in this study, it is observed that 1 and 2 hours delaying results in no difference if the initializations do not change. Moreover, more speed decreases and longer time intervals should be applied to individuals in order to get longer remaining distances in GA. Of course, the time delay for the remaining distance depends on the speed to be chosen.

5 Conclusion And Future Work

5.1 Conclusion

This study aimed to explore fuel consumption efficiency potential by speed optimization for vessels. Wind speed is observed as highly correlated with fuel consumption due to limited range of data source. So the wind speed parameter is excluded from analysis. By analyzing linear and non-linear machine learning regression models for fuel consumption prediction, this thesis has shown how fuel consumption can be predictable by using various environmental, operational and voyage real data logs of vessels. LR, SVR and KNN fuel consumption models resulted in lowest training accuracy, respectively, below 90%. On contrary, gradient boosting-based prediction models have shown the highest accuracies but highly included overfitting. VR ensemble model has resulted less overfitting in comparison of boosting models and it is selected for speed optimization. In this study, genetic algorithm showed good performance for random searching of speed profiles. This study is also performed the random speed initialization of genetic algorithms for speed optimization of ships which is not well studied in literature. Although some random distributions are used in recent studies for stochastic terms of GA individuals, our approach improves the searching capability of GA for the best individual i.e., speed profile for minimum fuel consumption during voyages. High crossover rates, population sizes, speed limits and low mutation rates observed better performance results in GA. However, the speed limits depend on the preferred voyage speed used in GA. So, defining speed limits without any consideration of voyage speed profile can cause less fuel consumption results with GA.

5.2 Future Work

Speed optimization with GA can be more efficient with multiple parameter optimization using random selection. Using also port delays, weather forecasts, wave conditions, bunker and fuel prices etc., in GA objective function can result more realistic results. In addition, using yearly data with more ship operational parameters and early stopping techniques can result in better accuracies in ML models.

6 Appendices

6.1 Appendix A: Method: Scikit-learn LocalOutlierFactor

Input Parameters	Options	Default Value	Description
n_neighbors	integer	<i>overridden</i>	<i>Number of neighbors</i>
algorithm	'auto', 'ball_tree', 'kd_tree', 'brute'	auto	<i>Algorithm used to compute the nearest neighbors</i>
metric	'cityblock', 'cosine', 'euclidean', 'l1', 'l2', 'manhattan', 'minkowski'	minkowski	<i>The metric is used for distance computation</i>
p	integer	2 (Euclidean distance)	<i>Parameter for the Minkowski metric</i>
contamination	'auto' or float	0.05	<i>The proportion of outliers in the data set</i>

7 References

1. Faber, J., et al., *Fourth IMO GHG Study*. International Maritime Organization, London, UK, 2020.
2. *2016 Guidelines for the development of a Ship Energy Efficiency Management Plan (SEEMP)*. Resolution MEPC.282(70).
3. IMO. *Energy Efficiency Measures*. 2022-05-06]; Available from: <https://www.imo.org/en/OurWork/Environment/Pages/Technical-and-Operational-Measures.aspx>.
4. IMO. *Data collection system for fuel oil consumption of ships*. 2022-05-06]; Available from: <https://www.imo.org/en/OurWork/Environment/Pages/Data-Collection-System.aspx>.
5. Faber, J., et al., *Regulated Slow Steaming in Maritime Transport. An assessment of options, costs and benefits*. 2012.
6. förstudie initierad av Lighthouse, E., *Consequences of speed reductions for ships*.
7. Psaraftis, H., *Speed Optimization vs Speed Reduction: the Choice between Speed Limits and a Bunker Levy*. Sustainability, 2019. **11**: p. 2249.
8. Arslan, O., E. Besikci, and A. Olcer, *Improving energy efficiency of ships through optimisation of ship operations*. No. FY2014-3 IAMU, 2014.
9. Aldous, L.G., *Ship operational efficiency: performance models and uncertainty analysis*. 2016, UCL (University College London).
10. Roar, A., C. Pierre, and W. Francois-Charles, *Optimal ship speed and the cubic law revisited: Empirical evidence from an oil tanker fleet*. Transportation Research Part E: Logistics and Transportation Review, 2020. **140**: p. 101972.
11. Kim, Y.-R., M. Jung, and J.-B. Park, *Development of a Fuel Consumption Prediction Model Based on Machine Learning Using Ship In-Service Data*. Journal of Marine Science and Engineering, 2021. **9**(2): p. 137.
12. Andrea, C., et al., *Vessels fuel consumption forecast and trim optimisation: A data analytics perspective*. Ocean Engineering, 2017. **130**: p. 351-370.

13. Christos, G., L. Iraklis, and T. Gerasimos, *Machine learning models for predicting ship main engine Fuel Oil Consumption: A comparative study*. Ocean Engineering, 2019. **188**: p. 106282.
14. Yang, L., et al., *Ship Speed Optimization Considering Ocean Currents to Enhance Environmental Sustainability in Maritime Shipping*. Sustainability, 2020. **12**(9): p. 3649.
15. Aydin, N., H. Lee, and S.A. Mansouri, *Speed optimization and bunkering in liner shipping in the presence of uncertain service times and time windows at ports*. European Journal of Operational Research, 2017. **259**(1): p. 143-154.
16. Helong, W., L. Xiao, and M. Wengang, *Voyage optimization combining genetic algorithm and dynamic programming for fuel/emissions reduction*. Transportation Research Part D: Transport and Environment, 2021. **90**: p. 102670.
17. Karthe, *Going Deeper into Regression Analysis with Assumptions, Plots & Solutions*. 2016.
18. Yang, L. and A. Shami, *On hyperparameter optimization of machine learning algorithms: Theory and practice*. Neurocomputing, 2020. **415**: p. 295-316.
19. Vorkapić, A., R. Radonja, and S. Martinčić-Ipšić, *Predicting Seagoing Ship Energy Efficiency from the Operational Data*. Sensors, 2021. **21**(8): p. 2832.
20. Kelleher, J.D., B. Mac Namee, and A. D'arcy, *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. 2020: MIT press.
21. Uyanık, T., Ç. Karatuğ, and Y. Arslanoğlu, *Machine learning approach to ship fuel consumption: A case of container vessel*. Transportation Research Part D: Transport and Environment, 2020. **84**: p. 102389.
22. Pedregosa, F., et al., *Scikit-learn: Machine learning in Python*. the Journal of machine Learning research, 2011. **12**: p. 2825-2830.
23. De Andrade, L.C.O., *Genetic Algorithms Application In Line Simplification*. 2014.
24. Johansson, C. and G. Evertsson, *Optimizing genetic algorithms for time critical problems*. 2003.
25. Chaal, M., *Ship operational performance modelling for voyage optimization through fuel consumption minimization*, in *Maritime Affairs*. 2018, World Maritime University: Malmö, Sweden.
26. Changnan, W., et al. *Optimization Analysis of USV Based on Genetic Algorithm*. in *Proceedings of the 2017 5th International Conference on*

Mechatronics, Materials, Chemistry and Computer Engineering (ICMMCCE 2017). 2017. Atlantis Press.

27. Hüffmeier, J., J. Lundman, and F. Elern, *Trim and Ballast Optimisation for a Tanker Based on Machine Learning*. 2020.
28. Xiao, C., *Using machine learning for exploratory data analysis and predictive models on large datasets*, in *Computer science*. 2015, University of Stavanger: Norway.
29. Kang, H., *The prevention and handling of the missing data*. Korean journal of anesthesiology, 2013. **64**(5): p. 402-406.
30. Blázquez-García, A., et al., *A review on outlier/anomaly detection in time series data*. arXiv preprint arXiv:2002.04236, 2020.
31. Breunig, M.M., et al., *LOF: identifying density-based local outliers*. SIGMOD Rec., 2000. **29**(2): p. 93–104.
32. Abebe, M., et al., *Machine Learning Approaches for Ship Speed Prediction towards Energy Efficient Shipping*. Applied Sciences, 2020. **10**(7): p. 2325.
33. Holmes, A., et al., *The Correlation Coefficient r* , in *STAT 462*. 2022.
34. Robert, N. *Regression diagnostics: Testing the assumptions of linear regression*. Statistical forecasting: Notes on regression and time series analysis 2022 [1/23/2022]; Available from: <https://people.duke.edu/~rnau/testing.htm>.
35. Kee, K.K., S.L. Boungh Yew, and K.-H. Renco, *Prediction of Ship Fuel Consumption and Speed Curve by Using Statistical Method*. 2018.
36. Hu, Z., et al., *A Novel Hybrid Fuel Consumption Prediction Model for Ocean-Going Container Ships Based on Sensor Data*. Journal of Marine Science and Engineering, 2021. **9**(4): p. 449.
37. Mosayebi, M. and M. Sodhi, *Tuning genetic algorithm parameters using design of experiments*. 2020: p. 1937-1944.