Determining residuary resistance per unit weight of displacement with Symbolic Regression and Gradient Boosted Tree algorithms

Baressi Šegota, Sandi; Lorencin, Ivan; Šercer, Mario; Car, Zlatan

Source / Izvornik: **Pomorstvo, 2021, 35, 275 - 284**

Journal article, Published version
Rad u časopisu, Objavljena verzija rada (izdavačev PDF)

https://doi.org/10.31217/p.35.2.11

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:187:446153

Multidisciplinary
SCIENTIFIC JOURNAL
OF MARITIME RESEARCH

University of Rijeka
FACULTY OF MARITIME STUDIES

Multidisciplinarni
znanstveni časopis
POMORSTVO

https://doi.org/10.31217/p.35.2.11

Determining residuary resistance per unit weight of displacement with Symbolic Regression and Gradient Boosted Tree algorithms

Sandi Baressi Šegota[1], Ivan Lorencin[1], Mario Šercer[2], Zlatan Car[1]

[1] University of Rijeka, Faculty of Engineering, Vukovarska 58, 51000 Rijeka, Croatia, e-mail: sbaressisegota@riteh.hr; ilorencin@riteh.hr; car@riteh.hr
[2] Razvojno edukacijski centar za metalsku industriju Metalska jezgra Čakovec, Bana Josipa Jelačića 22 D, 40000 Čakovec, Croatia, e-mail: mario.sercer@mev.hr

ABSTRACT

Determining the residuary resistance per unit weight of displacement is one of the key factors in the design of vessels. In this paper, the authors utilize two novel methods – Symbolic Regression (SR) and Gradient Boosted Trees (GBT) to achieve a model which can be used to calculate the value of residuary resistance per unit weight, of displacement from the longitudinal position of the center of buoyancy, prismatic coefficient, length-displacement ratio, beam-draught ratio, length-beam ratio, and Froude number. This data is given as results of 308 experiments provided as a part of a publicly available dataset. The results are evaluated using the coefficient of determination (R^2) and Mean Absolute Percentage Error (MAPE). Pre-processing, in the shape of correlation analysis combined with variable elimination and variable scaling, is applied to the dataset. The results show that while both methods achieve regression results, the result of regression of SR is relatively poor in comparison to GBT. Both methods provide slightly poorer, but comparable results to previous research focussing on the use of "black-box" methods, such as neural networks. The elimination of variables does not show a high influence on the modeling performance in the presented case, while variable scaling does achieve better results compared to the models trained with the non-scaled dataset.

ARTICLE INFO

Original scientific paper
Received 15 October 2021
Accepted 2 November 2021

Key words:
Artificial Intelligence
Gradient Boosted Trees
Hydrodynamic Modelling
Machine Learning
Symbolic Regression

1 Introduction

Artificial Intelligence is a commonly used tool in today's scientific and engineering practice, as its' modeling capabilities are extremely high, and may allow for the creation of high-precision models for many complex problems [1]. These techniques have been applied in many areas of maritime research. Examples include optimization of exergy analysis of internal ship systems [2] for steam turbines [3], modeling of propulsion system parameters [4, 5], ship modeling [6], vessel route optimization [7], and vessel detection [8].

There are many more examples of artificial intelligence applications. Oslebo et al. (2020) [9] apply machine learning for fault detection of pulsed-energy mission loads. Authors address the issue of discerning faults from the sudden heavy loads commonly present during the operation. Through the classification using the proposed novel machine learning method, the authors manage to achieve 99.8% accuracy in waveform classification and 100% accuracy in general fault detection. Berghout et al. (2021) [10] demonstrate a supervised deep learning approach for addressing the problem of condition-based maintenance of naval propulsion systems. Authors manage to achieve highly precise models through the application of the so-called extreme learning machine. Jeong et al. (2020) [11] demonstrate the application of machine learning for shipbuilding master data management. The authors demonstrate how machine learning can be applied to address the problem of an ever-increasing amount of data present in modern shipbuilding. Shaeffer et al. (2020) [12] apply machine learning in early-stage hull form design. Authors demonstrate that, as in many other industries, shipbuilding can apply data-driven models for determining the basic parameters of the hull forms. Barua et al. (2020) [13] review applications of machine learning for the problem of international freight transportation management. Authors review the most successful approaches and conclude that

the development of this kind of system should continue, as their uses are highly beneficial. In this paper the authors will consider the possibility of applying machine learning techniques on the modeling of yacht hydrodynamics, specifically modeling of residuary resistance per unit weight of displacement. Previous approaches have been made using neural networks [14] such as multilayer perceptron [15]. As both of those methods are so-called "black-box" methods, which experience the issue of inexplicability. The high complexity of these models does not allow for the interpretation of the models. Additionally, the neural network models tend to require a specific programming language, or even a specific library, to be re-used and applied. The methods that authors apply – GBT and Symbolic Regression are explainable methods that have equation and tree-shaped models, respectively. This allows them to more easily be implemented in various tools, without requiring specific function libraries. The novelty of this paper lies in the determination of the usability of the two used methods.

In the paper, first, the used dataset will be presented, followed by brief descriptions of methods, and used machine learning methodology. Finally, results will be presented and discussed, with conclusions drawn.

2 Methodology

The utilized methods are presented in this section. First, the analysis of the dataset is presented – using correlation and distribution analysis.

2.1 Dataset

The used dataset is the Delft yacht hydrodynamics data set, which was collected at the Delft Ship Hydromechanics Laboratory [16]. It consists of 308 full-scale experiments with 22 different hull forms. The dataset consists of 6 input variables and one input variable. The input variables are:

- The longitudinal position of the center of buoyancy
- Prismatic coefficient,
- Length-displacement ratio
- Beam-draught ratio
- Length-beam ratio, and
- Froude number.

All the input variables, as well as the output variable, are adimensional. The output variables describe the residuary resistance per unit weight of displacement, which determines the resistance a ship hull form experiences in regards to the displacement of the hull [16].

Before machine learning is applied, three analyses are performed – first is the determination of standard statistical descriptors, second is the distribution determinators, and finally the correlation analysis. Standard statistical descriptors are calculated for each variable and include the minimal and maximal values of the variable, range of the variable, the median value of the variable, and standard deviation of the median. The distribution of the variables is then plotted for each of the variables. This is achieved by plotting histograms of the variables and is used to determine if there are any outliers in the data that may cause issues in the creation of the regression models. Finally, correlation analysis is performed to determine which variables influence the output of the dataset. This can be useful for eliminating the variables which do not have a high influence on the input, which may assist with easier model regression [17].

Table 1 shows the standard statistical measures for each variable. We can see that each variable has a different range, which can negatively affect the performance of the used regression algorithms [18]. It should be also noted that the standard deviations, when compared to median value and range of the variable, are relatively low – except in the case of our output, residuary resistance per unit weight of displacement. This can indicate that the data in question has a relatively uneven distribution across its range, meaning that more data points are located on one end of the range [19, 20]. This can be confirmed by viewing the distributions of the variables, by plotting the histograms of the data, which is shown in Figure 1.

Figure 1 demonstrates the distribution of each input variable contained in the dataset, while Figure 2 shows the distribution of the output variable. It can be seen that

Table 1 The statistical descriptors of the variables in the dataset

	The longitudinal position of the center of buoyancy	Prismatic coefficient	Length-displacement ratio	Beam-draught ratio	Length-beam ratio	Froude number	Residuary resistance per unit weight of displacement
MIN	-5.00	0.53	4.34	2.81	2.73	0.13	0.01
MAX	0.00	0.60	5.14	5.35	3.64	0.45	62.42
RANGE	5.00	0.07	0.80	2.54	0.91	0.33	62.41
MEDIAN	-2.30	0.57	4.78	3.96	3.15	0.29	3.07
DEVIANCE	1.51	0.02	0.25	0.55	0.25	0.10	15.16

Source: Authors

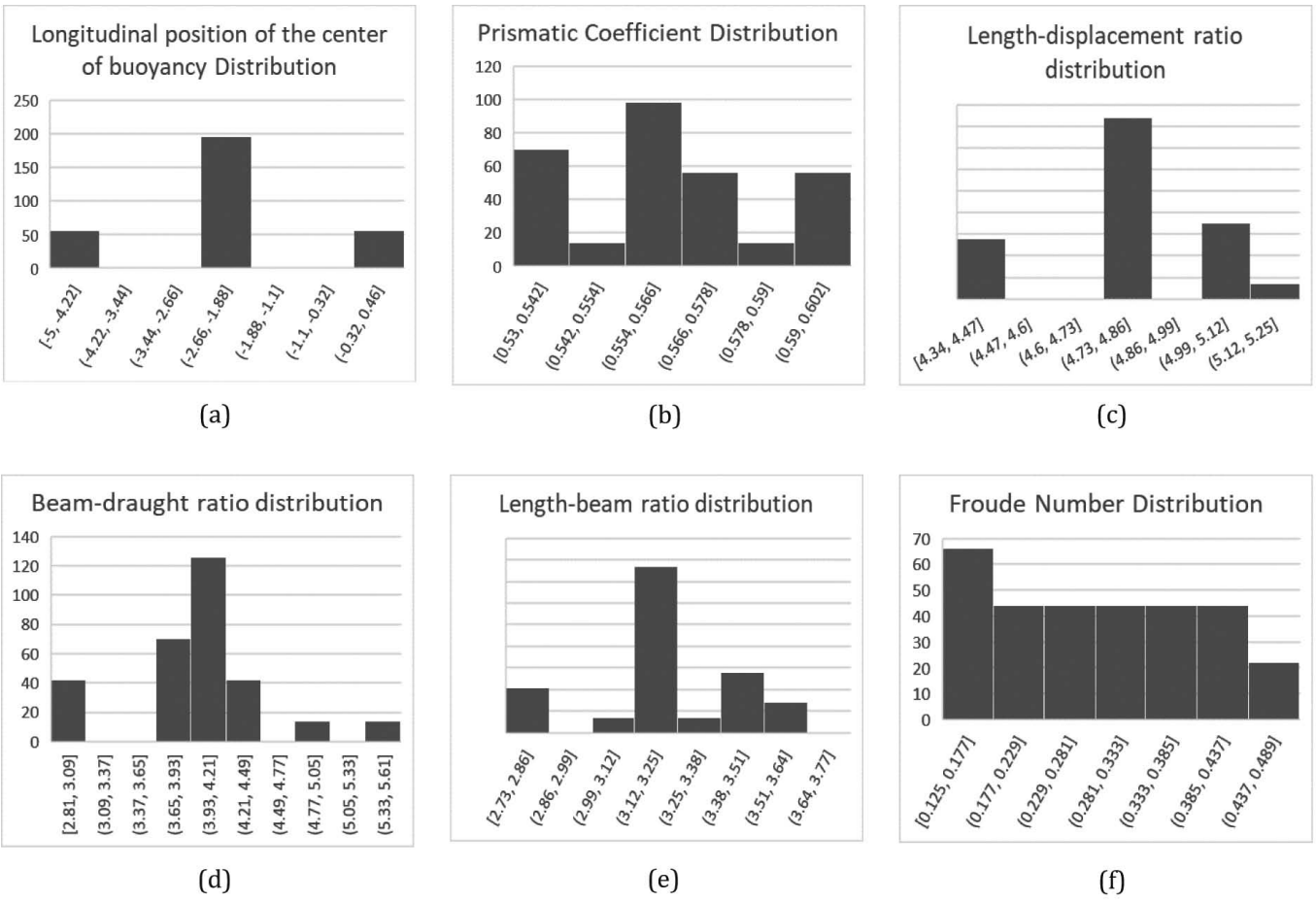

Figure 1 The distribution of variables in the dataset for (a) longitudinal position of the center of buoyancy, (b) prismatic coefficient, (c) length-displacement ratio, (d) beam-draught ratio, (e) length-beam ratio, and (f) Froude number.

Source: Authors

all of the input variables have similarities to the normal or uniform distributions. This is a good quality, which is commonly wanted within datasets used in machine learning applications [21]. The output is distributed exponentially, as shown in Figure 2. meaning that a larger amount of data is contained at the lower ranges of the dataset. This can cause issues with the models being better fitted for that data, as opposed to the general data [22]. Another element of note when observing Figures 1 and 2 is that the data is continuously distributed across each of the histogram bins, signifying that there are no outliers contained within data of each variable, meaning that the analysis and removal of those values are unnecessary.

The final performed analysis is the correlation analysis. Correlation analysis provides information on the inter-influence of individual variables within the dataset on one another. If x and y represent two datasets of length n for which we are trying to determine the correlation, then the correlation coefficient r is calculated according to equation [23, 24]:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})\sum_{i=1}^{n}(y_i - \bar{y})}} \quad (1)$$

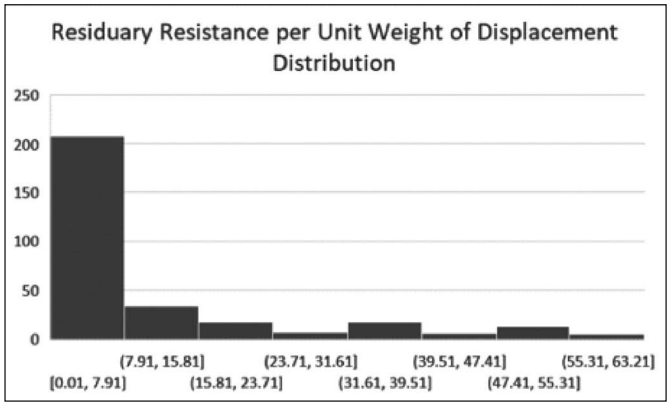

Figure 2 The distribution of output, residuary resistance per unit weight of displacement distribution

Source: Authors

The values for each variable achieved using the described methodology are given in Table 2. The table in question can allow us to perform variable elimination on the dataset to allow for easier regression using machine learning methods used in the presented work. The dataset is finally split into two subsets – the training and the

Table 2 The correlation between the variables in the dataset

	The longitudinal position of the center of buoyancy	Prismatic coefficient	Length-displacement ratio	Beam-draught ratio	Length-beam ratio	Froude number	Residuary resistance per unit weight of displacement
The longitudinal position of the center of buoyancy	1.00	-0.01	0.00	0.00	0.00	0.00	0.02
Prismatic coefficient	-0.01	1.00	-0.05	0.34	-0.09	0.00	-0.03
Length-displacement ratio	0.00	-0.05	1.00	0.38	0.68	0.00	0.00
Beam-draught ratio	0.00	0.34	0.38	1.00	-0.38	0.00	-0.01
Length-beam ratio	0.00	-0.09	0.68	-0.38	1.00	0.81	0.00
Froude number	0.00	0.00	0.00	0.00	0.00	1.00	0.81
Residuary resistance per unit weight of displacement	0.02	-0.03	0.00	-0.01	0.00	0.81	1.00

Source: Authors

testing set. The training set is used for the training of the regression models, while the testing set represents the previously unseen data for the models. This unseen data is used to evaluate the models, according to the metrics described in the following sections. In the presented research, the dataset was split into a 90:10 train/test ratio. This means that 277 data points have been used for the training part of the dataset, and 31 data points have been used for the testing part of the dataset.

The first variable is the longitudinal position of the center of buoyancy, which describes the position of the buoyancy center to the length of the vessel. The second variable is the prismatic coefficient which describes the distribution of displacement along a hull. The third input variable is the length-displacement ratio which describes the proportion of the vessel length and its displacement, while the fourth input – the beam-draught ratio describes the relationship between the waterline beam and the vessel draft. The fifth input variable is the length-beam ratio which describes the proportion between the vessel length and beams. The final input variable is the Froude number which defines the ratio of the flow inertia to the external field. The output, residuary resistance per unit weight of displacement describes the amount of resistance experienced in the dependence with vessel displacement expressed in unit weights [15].

2.2 AI regression

In this section the pre-processing applied to the data is described, followed by a brief overview of the used methods and their application. Finally, the method evaluation metrics are given.

2.2.1 Pre-processing

Two types of pre-processing are applied to the dataset in an attempt to improve the results. These are the scaling of the values in the dataset and the elimination of values that show a poor correlation to the output variable.

The min-max scaling is performed by taking the maximal and minimal values of each variable (given in Table 1) and transforming the value based on them to set the range of the variable to [0,1]. For a variable x, transformation into the scaled variable x' is done with [25, 26]:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2}$$

The variable elimination process is based on the values of the correlation coefficient between variables, as seen in Table 2. Due to the low correlation of most input variables with the output variable (observe last row or column), the elimination will be done in two cases. First, only variables with $|r| \geq 0.05$ will be kept, while in the second, the variables with $|r| \geq 0.01$ will be kept. Observing Table 2, it can be noted that for the first elimination only the Froude number will be kept, while in the second case the variables Froude number, beam-draught ratio, prismatic coefficient, and the length-beam ratio will be kept and used in the regression modeling.

One of the common steps in the data science application is the use of cross-validation, in which the data is split into multiple folds, and the training is performed multiple times with each of the folds being used for the training [27]. The reason this process has not been applied in the presented research is two-fold. The first reason is that the data shows distributions that are close to the normal

or uniform distributions, considering the small amount of data. The second is to allow for a more direct comparison to the previous research in the application of machine learning methods on the dataset used in this research – as the research in question has not used the cross-validation method, but instead used the standard train-test data split [15, 16], used in the presented research.

2.2.2 Random Search procedure

The selection of hyperparameters for both methods is performed through a random search. This means that the selection of each hyperparameter is performed uniformly randomly across the given range, the model is trained with the randomly selected parameters and the quality of the trained model is evaluated. This process is then repeated until either a satisfactory quality is achieved, or the execution is terminated due to the number of iterations elapsed reaching the pre-set value, which was set to 500 in the presented research. The selection of this procedure, as opposed to more strictly defined hyperparameter searches such as grid search or similar, was done due to the nature of the algorithms used. As opposed to algorithms such as neural networks which may have a highly discrete hyperparameter space, the hyperparameter space of the SR and GBT are more finely granular, which means that comparatively, small hyperparameter changes could result in significant model performance changes [27, 28]. The hyperparameter values used as bounding ranges for individual hyperparameters have been selected according to previous research and common practices [27–31].

2.2.3 Symbolic Regression

SR, also known as Genetic Programming (GP), is a method that utilizes the principles of evolutionary computing to develop regression models [27, 28]. SR creates the initial set of random solutions, called population, formatted as tree-shaped equations. The fitness of each of the solutions is determined. This means that the quality of each solution is ascertained, according to how well it models the data. Then, three different operations are applied to this set of random solutions – crossover, mutation, and reproduction. Crossover is applied to two separate candidate solutions. The tree-shaped equations are split and the new, child solutions, are accomplished via the recombination of the split parts. The equations for crossover operation are selected with the probability proportional to their fitness. This means that the probability of being selected for crossover and producing a child solution is higher for better solutions. Applying this operation repeatedly should, in theory, by combining the higher-quality solutions lead to better solutions being found from one generation to the next [29, 30]. Still, just the crossover application may cause some issues – such as narrowing the solution space search area and converging into a locally optimal solution [31, 32]. For this reason, two other operations are applied. The mutation will randomly modify a single, randomly selected solution – and include it into the next population iteration. The modification will either be done to a single node of the solution (point mutation), the subtree of the solution (subtree mutation), or through the subtree removal (hoist mutation). The reproduction will in turn simply copy an existing solution into the next population iteration to guarantee the gene pool health [37]. The solution selected for reproduction won't be selected fully randomly, but proportionally to the fitness. The probabilities of each of these operations being performed are the key hyperparameters of the SR method. The ranges of the hyperparameters used are given in Table 3.

In addition to evolutionary operations probabilities described previously, hyperparameters include population size, which is the initial set of the solutions, number of generations that controls the number of iterations in which the evolutionary applications are applied, and initial tree depth, which describes the maximal size of the equation trees in the initial population. It can be noted that the evolutionary operations are derived from the probabilities of other operations. This is done because those operations probabilities need to add up to 1, as one of the operations needs to be performed in each of the iterations to allow for the models to achieve better regression [38].

Table 3 Ranges of hyperparameters used for the random hyperparameter search of SR

Hyperparameter	Symbol	Minimum	Maximum
Crossover probability	P_c	0.8	0.95
Point Mutation probability	P_{mp}	0.01	$1 - P_c$
Hoist Mutation probability	P_{mh}	0.01	$1 - (P_c + P_{mp})$
Subtree Mutation probability	P_{ms}	0.01	$1 - (P_c + P_{mp} + P_{mh})$
Reproduction probability	P_R	$1 - (P_c + P_{mp} + P_{mh} + P_{\_ms})$	$1 - (P_c + P_{mp} + P_{mh} + P_{\_ms})$
Population	$\mathbb{P}$	100	1000
Generations	G	200	1000
Initial tree depth	T_{id}	7	16

Source: Authors

2.2.4 GBT

GBT is a tree-based AI ensemble method [39]. It is also based on trees such as SR, but instead of those trees describing equations, they describe decision paths [40]. Each node of the tree describes a split into two paths depending on the value of a parameter, which leads down to tree leaves that contain regressed values. GBT being an ensemble method means it uses a voting system, in which many trees are generated, and the output value of the model is calculated not based on a single tree but as a weighted average of all trees in the ensemble. Gradient boosting is a process in which the models are trained based on the residual error of the models. The error is calculated in each of the iterations and the gradient is adjusted based on it. The training speed is then adjusted proportionally to the error [41]. This approach allows faster model convergence and avoids the problem of skipping the possible optimal solutions due to the training process slowing down when solutions near optimum are found [42].

Random search is also applied for the hyperparameters of GBT, with the possible values given in Table 4.

Across the hyperparameters number of estimators represent the number of tree models included in the ensemble. Maximum features describe the algorithm used to calculate the maximum number of nodes depending on tree depth, the maximum of which is contained in the hyperparameter value Maximal Tree Depth. Minimal samples for leaf and split describe the minimal number of data points needed for the creation of tree split or leaf within the model Finally, the training algorithms describe the algorithm used for the calculation of gradients [43].

2.2.5 Quality determination

Quality determination has been performed using two metrics – coefficient of determination (R^2) and Mean Absolute Percentage Error (MAPE). Both compare the real dataset values y to the set of predicted dataset values $\hat{y}$. R^2 is calculated according to the equation [40, 41]:

$$R^2 = 1 - \frac{\sum_{i=0}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n}\left(y_i - \frac{1}{n}\sum_{i=0}^{n} y_i\right)^2} \tag{3}$$

Using the same notation, MAPE is calculated using the equation [46]:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{4}$$

where n is the number of elements in the test set. R^2 is an adimensional value that defines the amount of variance from the real data that is contained within the predicted data [43, 44]. This value is commonly used in the evaluation of regression models, as it provides a good description of how well the model reacts to variation of the output variable in the real dataset. MAPE in turn is given as the percentage of the range of the variable. It describes the average error the model achieves across the test set. The benefit of MAPE is that, as the error is given as a percentage, it is easy to interpret and understand its value. Beyond that, its performance is extremely similar to more commonly used MAE [48].

3 Results and Discussion

Figures 2 and 3 show the results achieved by the algorithms. In both figures, the scores are given for both GBT and SR algorithms, across all possible variations of dataset pre-processing (scaling and variable elimination).

As it can be noticed from Figure 2 the GBT achieves higher R^2 scores than SR. The highest R^2 score is achieved by GBT with data scaling applied, regardless of the variable elimination criterion applied. Higher R^2 scores are achieved on the scaled data in both algorithms, while variable elimination does show somewhat improved R^2 scores, although those differences are not as visible as with data scaling.

The MAPE scores are presented in the same manner as the R^2 scores. The best error achieved, of 1.48%, corresponds to the highest R^2 score, as it is achieved by the GBT method on scaled data, regardless of the variable elimination correlation criterion.

Tables 5 and 6 represents hyperparameters that were used by the best solutions. It has to be noted that many solutions achieved similar scores, as minor hyperparameter variations can lead to extremely similar models which may

Table 4 Ranges of hyperparameters used for the random hyperparameter search of GBT

Hyperparameter	Minimum Value	Maximal Value
Number of Estimators	10	50
Maximum Features	Automatic, Square Root, Base-2 Logarithm	
Maximal Tree Depth	10	30
Minimal Leaf	2	50
Minimal Split	2	50
Training Algorithm	Gbtree, Dart	

Source: Authors

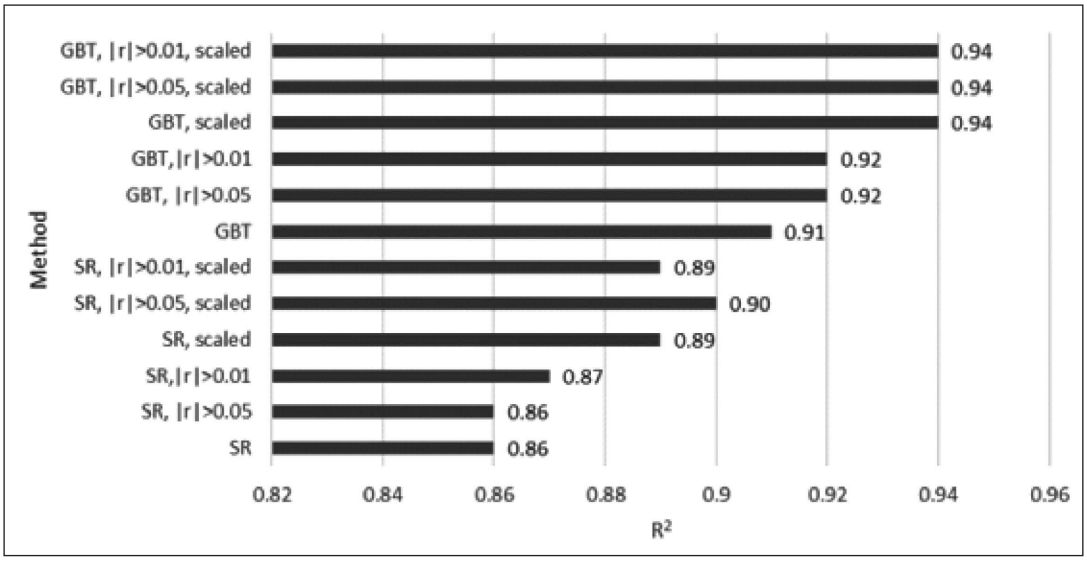

Figure 3 Comparison of the results across all methods and variations used via R^2 (Higher is better)

Source: Authors

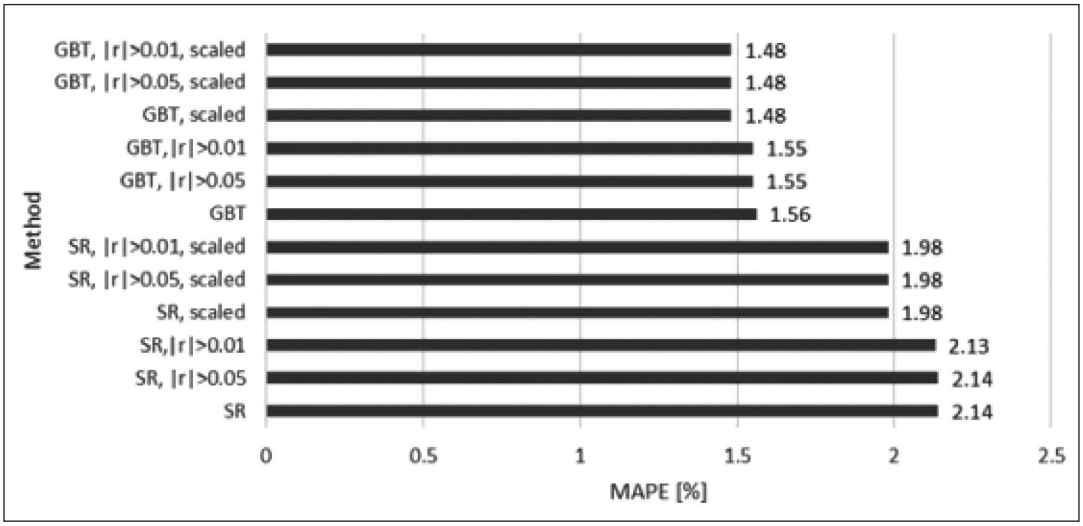

Figure 4 Comparison of the results across all methods and variations used via MAPE (Lower is better)

Source: Authors

Table 5 Hyperparameter models of best models per variation for SR

	Crossover probability	Point Mutation probability	Hoist Mutation probability	Subtree Mutation probability	Reproduction probability	Population	Generations	Initial tree depth
SR	0.91	0.03	0.02	0.01	0.03	923	892	12
SR, $\|r\|$>0.05	0.90	0.04	0.04	0.01	0.01	814	913	15
SR, $\|r\|$>0.01	0.92	0.02	0.01	0.04	0.01	987	968	15
SR, scaled	0.91	0.03	0.02	0.02	0.03	940	898	14
SR, $\|r\|$>0.05, scaled	0.94	0.01	0.02	0.03	0.00	914	912	16
SR, $\|r\|$>0.01, scaled	0.91	0.03	0.01	0.03	0.02	992	997	10

Source: Authors

Table 6 Hyperparameter models of best models per variation for GBT

	Number of Estimators	Maximum Features	Maximal Tree Depth	Minimal leaf Samples	Minimal Split Samples	Training Algorithm
GBT	48	Square Root	29	14	24	GBTree
GBT, $\|r\|>0.05$	42	Square Root	27	6	23	GBTree
GBT, $\|r\|>0.01$	49	Square Root	22	12	17	GBTree
GBT, scaled	44	Square Root	25	29	39	GBTree
GBT, $\|r\|>0.05$, scaled	45	Square Root	24	4	48	GBTree
GBT, $\|r\|>0.01$, scaled	48	Square Root	24	25	3	GBTree

Source: Authors

result in the same, or extremely similar, scores – especially on smaller datasets such as the one used in this research. For brevity, only the absolute best hyperparameter combinations are presented. In addition, in the case where two of the solutions achieved equal R^2 scores the one with the lower MAPE was selected as the better solution. If two or more solutions achieved the same scores when rounded to the 9th decimal, the presented solution was selected as one with lower complexity (lower number of tree models for GBT, smaller population in SR), as those solutions are easier to train. The solutions presented in the Table correspond to the ones as in Figures 2 and 3, so the achieved scores have not been repeated.

Observing the hyperparameters for best models for SR it can be noted that all models utilized relatively high crossover probability, equal to or higher than 0.9. High values have also been used for the population, generations, and initial tree depth. Using the higher range of hyperparameters for best solutions suggests that the regression problem is relatively hard.

From Table 6 it is noticeable that relatively high values are used for the number of estimators and maximal tree depth. As it was with SR, this also tends to suggest a relatively hard regression problem. Interestingly, all the best solutions use square root for calculating the maximum number of features, and GBTree for the training algorithm.

4 Conclusion

The results suggest that the SR and GBT can be used for regression of the residuary resistance per unit weight of distribution. Out of the two methods, the models regressed with GBT show a higher quality regression. While in comparison to previous research [15] the methods achieved are of a lower quality, the findings point out that both methods, especially GBT, could be used to address the presented, or similar, problems. Observing the hyperparameters selected during the training process, it can be noted that they tended towards the higher end of the hyperparameter range. This suggests that further increasing the hyperparameter range could be used to achieve better results.

Analysis of the dataset shows that the output values of the dataset, for the value of the residuary resistance per unit weight of displacement, are not uniformly or normally distributed which could be part of the issues causing the SR and GBT methods to fail to regress the problem with higher quality. In the use of neural networks, this can be addressed by using the large number of weighted neuron connections, which successfully make up for the lacking correlation information between the input and output variables. But, due to the lower complexity of SR and GBT methods, the same is not the case when regression is performed using them. Future research should focus on the analysis of the dataset and its composition, so it could be more easily determined if further statistical analytics and data pre-processing could be applied to reduce the problem complexity.

An important note in the dataset composition is that variable elimination does not show a significant lowering of the scores – even when a total of five out of six variables are removed. This confirms the performed correlation analysis and suggests that the Froude number may be a key factor in AI modeling of residuary resistance of the ship hull forms, while the other input variables from the dataset, especially length-beam and length-displacement ratios may not be necessary for modeling. Future work could focus on applying further validation on the variable-removed dataset, for example using k-fold cross-validation or similar approaches, to test the generalization properties of the models.

Finally, it can be concluded that explainable models can be used to solve relatively complex problems in the maritime environment and modeling, and such approaches should be given consideration by the researchers. Future work in the field may include the application of SR, GBT, or similar AI modeling techniques in the modeling of energy [49], environmental effects [50], or internal systems [51] within the domain of maritime applications.

Funding: The presented research has not been funded by external sources.

Acknowledgments: This research has been (partly) supported by the CEEPUS network CIII-HR-0108, European Regional Development Fund under the grant KK.01.1.1.01.0009 (DATACROSS), project CEKOM under

the grant KK.01.2.2.03.0004, CEI project COVIDAi (305.6019-20), project Metalska jezgra Čakovec (KK.01.1.1.02.0023) and University of Rijeka scientific grant uniri-tehnic-18-275-1447.

Author contributions: Conceptualization, S.B.Š, I.L., methodology, S.B.Š., M.Š., Z.C.; software, S.B.Š, I.L.; validation, M.Š., Z.C; formal analysis, M.Š., Z.C., investigation, S.B.Š., I.L., M.Š., Z.C., resources, M.Š., Z.C., data curation, S.B.Š., I.L., writing – original draft preparation, S.B.Š., I.L.; writing – review and editing, M.Š., Z.C.; visualization, S.B.Š.; supervision, M.Š., Z.C; project administration, M.Š., Z.C; finding acquisition, M.Š., Z.C; final approval, M.Š., Z.C.

References

[1] Z. Car, S. Baressi Šegota, N. Anđelić, I. Lorencin, and V. Mrzljak, "Modeling the Spread of COVID-19 Infection Using a Multilayer Perceptron", *Comput. Math. Methods Med.*, vol. 2020, 2020, doi: 10.1155/2020/5714714.

[2] V. Mrzljak, I. Poljak, J. Prpić-Oršić, and M. Jelić, "Exergy analysis of marine waste heat recovery CO2 closed-cycle gas turbine system", *Pomorstvo*, vol. 34, no. 2, pp. 309–322, 2020.

[3] S. Baressi Šegota, I. Lorencin, N. Anđelić, V. Mrzljak, and Z. Car, "Improvement of Marine Steam Turbine Conventional Exergy Analysis by Neural Network Application", *J. Mar. Sci. Eng.*, Vol. 8, No. 11, p. 884, 2020.

[4] N. Anđelić, S. Baressi Šegota, I. Lorencin, I. Poljak, V. Mrzljak, and Z. Car, "Use of Genetic Programming for the Estimation of CODLAG Propulsion System Parameters", *J. Mar. Sci. Eng.*, Vol. 9, No. 6, p. 612, 2021.

[5] I. Lorencin, N. Anđelić, V. Mrzljak, and Z. Car, "Multilayer perceptron approach to condition-based maintenance of marine CODLAG propulsion system components", *Pomorstvo*, Vol. 33, No. 2, pp. 181–190, 2019.

[6] P. D. Sclavounos and Y. Ma, "Artificial intelligence machine learning in marine hydrodynamics", in *International Conference on Offshore Mechanics and Arctic Engineering*, 2018, Vol. 51302, p. V009T13A028.

[7] H. Yu *et al.*, "Ship Path Optimization That Accounts for Geographical Traffic Characteristics to Increase Maritime Port Safety", *IEEE Trans. Intell. Transp. Syst.*, 2021.

[8] I. Lorencin, N. Anđelić, V. Mrzljak, and Z. Car, "Marine objects recognition using convolutional neural networks", *NAŠE MORE Znan. časopis za more i Pomor.*, Vol. 66, No. 3, pp. 112–119, 2019.

[9] D. Oslebo, K. A. Corzine, T. Weatherford, and A. Maqsood, "Fault Detection for Naval Pulsed-Energy Mission Loads Using a Novel Machine Learning Approach", *Nav. Eng. J.*, Vol. 133, No. 1, pp. 69–81, 2021.

[10] T. Berghout, L.-H. Mouss, T. Bentrcia, E. Elbouchikhi, and M. Benbouzid, "A deep supervised learning approach for condition-based maintenance of naval propulsion systems", *Ocean Eng.*, Vol. 221, p. 108525, 2021.

[11] J. H. Jeong, J. H. Woo, and J. Park, "Machine learning methodology for management of shipbuilding master data", *Int. J. Nav. Archit. Ocean Eng.*, Vol. 12, pp. 428–439, 2020.

[12] A. K. Shaeffer, W. Wilson, and C. Yang, "Application of Machine Learning to Early-Stage Hull Form Design", 2020.

[13] L. Barua, B. Zou, and Y. Zhou, "Machine learning for international freight transportation management: a comprehensive review", *Res. Transp. Bus. Manag.*, Vol. 34, p. 100453, 2020.

[14] I. Ortigosa, R. Lopez, and J. Garcia, "A neural networks approach to residuary resistance of sailing yachts prediction", 2007.

[15] S. Baressi Šegota, N. Anđelić, J. Kudláček, and R. Čep, "Artificial neural network for predicting values of residuary resistance per unit weight of displacement", *Pomor. Zb.*, Vol. 57, No. 1, pp. 9–22, 2019.

[16] J. Gerritsma, R. Onnink, and A. Versluis, "Geometry, resistance and stability of the delft systematic yacht hull series", *Int. Shipbuild. Prog.*, Vol. 28, pp. 276–297, 1981.

[17] H. Zhao, "Dynamic graph embedding for fault detection", *Comput. Chem. Eng.*, Vol. 117, pp. 359–371, Sep. 2018, doi: 10.1016/j.compchemeng.2018.05.018.

[18] M. Azizjon, A. Jumabek, and W. Kim, "1D CNN based network intrusion detection with normalization on imbalanced data", in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2020, pp. 218–224.

[19] S. T. Jagtap, K. Phasinam, T. Kassanuk, S. S. Jha, T. Ghosh, and C. M. Thakar, "Towards application of various machine learning techniques in agriculture", *Mater. Today Proc.*, 2021.

[20] Z. Liu, M. Qi, C. Shen, Y. Fang, and X. Zhao, "Cascade saccade machine learning network with hierarchical classes for traffic sign detection", *Sustain. Cities Soc.*, Vol. 67, p. 102700, 2021.

[21] I. Pavlova, D. Zikrach, D. Mosler, D. Ortenburger, T. Góra, and J. W\kasik, "Determinants of anxiety levels among young males in a threat of experiencing military conflict--Applying a machine-learning algorithm in a psychosociological study", *PLoS One*, Vol. 15, No. 10, p. e0239749, 2020.

[22] C. R. A. V. Oikawa, V. Freitas, M. Castro, and L. L. Pilla, "Adaptive load balancing based on machine learning for iterative parallel applications", in *2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 2020, pp. 94–101.

[23] A. Turkcan, "The effects of different types of biodiesels and biodiesel-bioethanol-diesel blends on the cyclic variations and correlation coefficient", *Fuel*, Vol. 261, p. 116453, 2020.

[24] M. E Ali and T. Medhat, "Correlation Coefficient Via Statistical and Rough Set Concepts", *Inf. Sci. Lett.*, Vol. 10, No. 3, p. 6, 2021.

[25] M. M. Ahsan, M. A. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance", *Technologies*, Vol. 9, No. 3, p. 52, 2021.

[26] A. Tripathi, N. Bhoj, M. Khari, and B. Pandey, "Feature Selection and Scaling for Random Forest Powered Malware Detection System", 2021.

[27] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[28] J. R. Koza and J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*, Vol. 1. MIT press, 1992.

[29] J. R. Koza, M. A. Keane, M. J. Streeter, W. Mydlowec, J. Yu, and G. Lanza, *Genetic programming IV: Routine human-competitive machine intelligence*, Vol. 5. Springer Science & Business Media, 2006.

[30] J. R. Koza and others, *Genetic programming II*, Vol. 17. MIT press Cambridge, 1994.

[31] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system", in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[32] T. Chen *et al.*, "Xgboost: extreme gradient boosting", *R Packag. version 0.4-2*, Vol. 1, No. 4, pp. 1–4, 2015.

[33] W. La Cava and J. H. Moore, "Learning feature spaces for regression with genetic programming", *Genet. Program. Evolvable Mach.*, Vol. 21, No. 3, pp. 433–467, 2020.

[34] T. Helmuth and A. Abdelhady, "Benchmarking parent selection for program synthesis by genetic programming", in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, 2020, pp. 237–238.

[35] A. V Grin and A. H. Gandomi, "Advancing Genetic Programming via Information Theory", in *2021 IEEE Congress on Evolutionary Computation (CEC)*, 2021, pp. 1991–1998.

[36] J. R. Koza, D. Andre, M. A. Keane, and F. H. Bennett III, *Genetic programming III: Darwinian invention and problem solving*, vol. 3. Morgan Kaufmann, 1999.

[37] A. Lalejini, M. A. Moreno, and C. Ofria, "Tag-based regulation of modules in genetic programming improves context-dependent problem solving", *Genet. Program. Evolvable Mach.*, Vol. 22, No. 3, pp. 325–355, 2021.

[38] M. Sipper and J. H. Moore, "Genetic Programming Theory and Practice: A fifteen-year trajectory", *Genet. Program. Evolvable Mach.*, Vol. 21, No. 1, pp. 169–179, 2020.

[39] S. Sachdeva and B. Kumar, "Comparison of gradient boosted decision trees and random forest for groundwater potential mapping in Dholpur (Rajasthan), India", *Stoch. Environ. Res. Risk Assess.*, Vol. 35, No. 2, pp. 287–306, 2021.

[40] Z. Zhang and C. Jung, "GBDT-MO: Gradient-Boosted Decision Trees for Multiple Outputs", *IEEE Trans. Neural Networks Learn. Syst.*, 2020.

[41] T. Sharma, P. Gupta, V. Nigam, and M. Goel, "Customer Churn Prediction in Telecommunications Using Gradient Boosted Trees", in *International Conference on Innovative Computing and Communications*, 2020, pp. 235–246.

[42] C. Aguilar-Palacios, S. Muñoz-Romero, and J. luis Rojo-Álvarez, "Cold-start promotional sales forecasting through gradient boosted-based contrastive explanations", *IEEE Access*, Vol. 8, pp. 137574–137586, 2020.

[43] Z. Qin *et al.*, "Are Neural Rankers still Outperformed by Gradient Boosted Decision Trees?", 2021.

[44] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation", *PeerJ Comput. Sci.*, Vol. 7, p. e623, 2021.

[45] H. Kang, "Sample size determination and power analysis using the G* Power software", *J. Educ. Eval. Health Prof.*, Vol. 18, 2021.

[46] A. De Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean absolute percentage error for regression models", *Neurocomputing*, Vol. 192, pp. 38–48, 2016.

[47] T. Jiali, L. Yuxiang, Y. U. Guangping, and L. I. U. Jian, "Configuration optimization of air cathode microbial fuel cell and its performance evaluation for rapid determination of BOD", *Chinese J. Environ. Eng.*, Vol. 15, No. 6, pp. 2155–2164, 2021.

[48] U. Khair, H. Fahmi, S. Al Hakim, and R. Rahim, "Forecasting error calculation with mean absolute deviation and mean absolute percentage error", in *Journal of Physics: Conference Series*, 2017, Vol. 930, No. 1, p. 12002.

[49] A. I. Ölçer, M. Kitada, D. Dalaklis, and F. Ballini, *Trends and challenges in maritime energy management*, Vol. 6, Springer, 2018.

[50] I. Gospić, I. Glavan, I. Poljak, and V. Mrzljak, "Energy, Economic and Environmental Effects of the Marine Diesel Engine Trigeneration Energy Systems", *J. Mar. Sci. Eng.*, Vol. 9, No. 7, p. 773, 2021.

[51] L. Kocijel, I. Poljak, Z. Car, and others, "Energy loss analysis at the gland seals of a marine turbo-generator steam turbine", *Teh. Glas.*, Vol. 14, No. 1, pp. 19–26, 2020.