Prediction of vessel propulsion power using machine learning on AIS data, ship performance measurements and weather data

IOP Conf. Series: Journal of Physics: Conf. Series **1357** (2019) 012038 doi:10.1088/1742-6596/1357/1/012038

Prediction of vessel propulsion power using machine learning on AIS data, ship performance measurements and weather data

Q Liang[1], **H A Tvete**[1], **H W Brinks**[1]

[1] Group Technology and Research Maritime, Digital Class and Services, DNV GL, Veritasveien 1, 1363 Høvik, Norway
Email: qin.liang@dnvgl.com

Abstract. The aim of this paper is to utilize AIS (Automatic Identification System) data, ship propulsion power measurements and weather data and apply different machine learning (traditional and deep learning) methods to develop improved models to predict ship propulsion power. The performance between different traditional machine learning methods and deep learning with different architectures were compared and discussed. Two scenarios were explored: 1) Training a machine learning model on one container ship significantly improved the predictions of power demand as compared to a physics-based model, with an R^2 score of 0.78 compared to 0.48. 2) A machine learning model was also trained on several container ships. This scenario, where the test data with other container ships was not included in the training dataset, also showed better prediction with machine learning than physics-based model, with an improvement in R^2 score from 0.69 to 0.85. However, the use of the trained machine learning model on other ship types showed varying results and especially when the vessel size differed significantly from vessels in the training data, data-driven models showed limitations. For vessels of similar size, however, machine learning for other ship types showed improvements.

1. Introduction

In 2018, the International Maritime Organization (IMO) adopted an initial strategy to reduce greenhouse gas (GHG) emissions from ships. The strategy aims at reducing total GHG emissions from international shipping, peaking as soon as possible, and to be reduced by at least 50% by 2050 as compared to 2008. A massive adaptation of alternative fuels and energy efficiency measures needs to happen to reach this goal.

Shipping is also facing the introduction of the global 0.5% sulfur cap in 2020, one of the most important regulatory changes in its current history [1]. Approximately 90% of the world's trade is moved by shipping, and more than 70,000 ships will be affected by this regulation. Stricter limits on sulfur emissions are already in place in Emission Control Areas (ECAs) around the world. As a result of the increased international attention to global warming and air pollution, different stakeholders are beginning to take actions to reduce local and global emissions. Emission inventory models, resting on position data in combination with ship technical data and engine characteristics, have provided valuable input into fleet-wide operation and corresponding emissions. Modelling of maritime fuel consumption, emissions and discharges to sea from the shipping industry are fundamental as input to

evaluate impacts on the environment, human health and climate, and to effectively assess what option are available to mitigate the impacts. The common denominator is that they rely on physical correlations.

In contrast to physics-based models, the primary objective of this paper is to evaluate the performance of different machine learning models' ability to accurately predict ship propulsion power from AIS-, ship characteristics- and weather data. The various machine learning models were benchmarked against ship propulsion power measurement data and existing physical AIS-based propulsion power models. Both traditional machine learning and deep learning methods have been investigated. In this study, different MLP (Multi-Layer Perceptron) models with different architectures were benchmarked and discussed.

With the development of computing power and availability of data, machine learning methods have gained popularity in many domains, especially for cases with huge amounts of variables and complex relationship between them. Deep learning algorithms are one promising avenue of machine learning for the automated extraction of complex data representations (features) at high levels of abstraction [2]. Such algorithms develop a layered, hierarchical architecture of learning and representing data, where higher-level (more abstract) features are defined in terms of lower-level (less abstract) features.
A 'grey-box' modelling approach for the simulation of ocean vessels operation has been proposed [3]. The result shows that the grey-box model, as well as the black-box model, was able to follow the operational data relatively well, whereas the white-box in this case failed. When influencing factors are not taken into account or the parameters are outside the range of the model, white-box approach may fail. Meaning, for the situations which are complex or difficult to model, data-driven methods provide an alternative approach. Predicting ship design power is typically derived by applying a 'sea margin' onto a reference 'calm water power'. This is of questionable accuracy as the techniques available to estimate these 'sea margins' are inaccurate [4]. It is extremely difficult to analyze ship performance data with environmental factors using the physics-based methods, when the nature of the regression relationship is not known a priori.

Petersen J.P. et al. [5] investigated and compared traditional methods of estimating ship propulsion efficiency to two statistical model approaches: Artificial Neural Networks (ANN) and Gaussian processes (GP). Both statistical (data-driven) models performed better than the traditional models, and the performance of the ANN was found slightly better than that of the GP in all of the tests in the paper. An ANN-based decision support system for improving ship energy efficiency was proposed in [6], and this study presented the closed form solution of fuel consumption based on trained ANN parameters (weights and biases) as a function of input parameters. The reason why this paper can do this is that it only adapted one single layer structure for the neural network.

Most of the above mentioned studies adapted ANN, with less focus on traditional machine learning methods like random forest, decision tree and support vector machine. In most of the scenarios, traditional machine learning methods could also provide good enough results.

Other neural networks, with more complex structure, like CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network), could also potentially be applied. RNN with LSTM (Long Short-Term Memory) is particularly suitable for time-series prediction but challenging to apply on data with unbalanced sampling rate. RNN has, however, become less popular the last years for several reasons. Firstly, RNN still has problems with vanishing gradients even with LSTM. Secondly, it is more difficult to train an RNN than other neural network models when long sequences are required, or the structure is complex. A temporal convolutional network (TCN) architecture for sequence modeling was proposed, which convincingly outperform baseline recurrent architectures across a broad range of sequence modeling tasks [7]. The deep learning methods are continuously under development. Hence, when selecting a machine learning method, it is not obvious which method and structure that will perform the best. For example, CNN was designed for image recognition, while a simplified version CNN-TCN got better result in time-series prediction compared to RNN. As a conclusion; the methods or architecture of neural network does not need to be complex as long as it fit well with the data and provide a balanced performance.

In previous studies, data volume was found to be quite limited, e.g. with 233 noon reports from one vessel [6] and measurements from one vessel only in [8]. This limitation prevents ANN from showing its full strengths with learning from the large amounts of data. In this study, two scenarios were investigated. For the scenario 1, one vessel with 75,000 data sets of AIS (position) data, fuel consumption measurements and weather data were used for training and testing. 70% of the data was used for training and 30% for testing. In scenario 2, 238 container vessels were selected with in total about 400,000 measurements. In addition to having more measurements, scenario 2 also cover container vessels with different sizes. The test of scenario 2 consisted of ships not used in the training data, and the predictive power within the same ship type was tested. Furthermore, the predictive power of the trained machine learning model for other ship types than container ships was tested for scenario 1.

2. Methods

Machine learning methods show strong abstraction and learning ability with a potential to predict vessel propulsion power based on ship position data, ship technical information and weather data without understanding the physical relationship between input features and output propulsion power. In this study, the data includes features which are quite sophisticated and difficult to analyze by physics-based methods. Machine learning methods fill the gap of this limitation. The prediction results from both traditional machine learning and deep learning methods were compared.

2.1. Traditional Machine learning methods

Decision trees are a non-parametric supervised learning method used for classification and regression. The algorithm aims to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The deeper the tree, the more complex decision rules will be generated to filter the model [9].

Random forest is a kind of supervised machine learning algorithm based on ensemble learning. Ensemble learning is an algorithm in which different types of an algorithm or the same algorithm are combined multiple times to form a more powerful model. A random forest is a meta-estimator that fits multiple decision trees on each subsample of a dataset and uses averages to improve prediction accuracy and control overfitting. After searching and calculation, it finally came up with a forest of trees, hence named “random forest” [10].

Support Vector Regression (SVR) is based on Support Vector Machine (SVM) and operates with similar principles. In the simple regression model, the algorithm tries to minimize the error rate. SVR tries to fit the error within a certain range, which is a space separated by the boundary lines. In SVM/SVR boundary lines create a margin and separate two classes. The hyper planes are between the separation lines. In SVR they represent the lines that help to predict the continuous value. The objective of SVR is to basically consider the points that are within the boundary line. The best fit line is the hyper plane that has maximum number of data points. The principle of SVM/SVR is to transform the data into a higher dimensional space to make it possible to perform the separation. The functions that are used to transform data are called kernel functions. SVM/SVR works well on small datasets, but requires knowledge to select appropriate kernel function and parameters [11].

2.2. Artificial neural network (deep learning)

Artificial Neural Network, also called deep learning, has become popular in recent years. The application of ANN is quite wide, e.g. mathematical prediction, image recognition and speech recognition. ANN consists of connected neurons distributed in several layers. Simple ANNs only contain one weight and bias for each neuron. Complex ANNs can have more parameters and a complex structure in each neuron. A simplified illustration of an ANN is presented in Figure 1.

ANN consists of an input layer, one or several hidden layers and an output layer. In each layer there are different number of neurons which are defined by the number of features. First, the neurons calculate a weighted sum of its input, then add a bias to it. Subsequently, the activation function is used and this ‘activated’ output will be transferred to the next layer for further calculation or output.

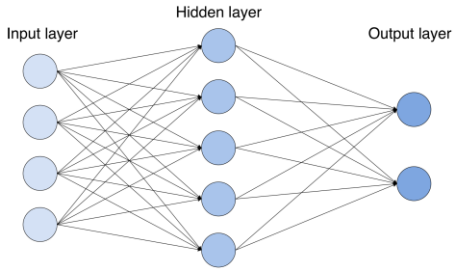

Figure 1. A simple neural network with one hidden layer

The depth of ANN defines the learning ability. This does not necessarily mean that deeper ANNs with many layers performs better than shallower ones. Shallow and wide ANNs can memorize data well but be worse at generalization. Deeper ANNs are capable of learning features at various abstraction allowing explicit development of areas of the network to handle the weaker relationships between inputs and outputs. This is beneficial to improve model generalization and enhance the ability to deduce the output based on input. If the ANN is deeper than required, this will result in the degradation problem (accuracy does not improve by adding more layers). Hence, appropriate depth and configuration of ANN with well-processed data sets will provide the best performance.

ANN is an advanced linear combination of nonlinear regression models that can be used to model complex relationships between input and output variables. There are different types of ANNs, including MLP, CNN and RNN, which fit for different applications. CNN is mainly utilized for image recognition. RNN can be used for time-series prediction and speed recognition, which can remember what happened in the past. However, there are no obvious boundaries between the applications of ANNs. E.g., there are researchers trying to apply CNN with a new structure for time-series prediction and the results are convincingly better than the RNN [7]. In this study, MLP with different structures will be adapted for the prediction.

2.3. Physics-based model

The physics-based model that is used for benchmarking purposes in this study, has been developed by DNV GL [12]. This model predicts the propulsion power as being proportional to the third order of the operational speed. By utilizing ship technical data and engine characteristics, the fuel consumption and corresponding emissions can be calculated. The main engine fuel consumption FC_x^{ME} associated with AIS position message x (in kg fuel) is calculated as following:

$$FC_x^{ME} = P^{ME} SFOC^{ME} LF_x^{ME} \Delta t_x \quad (1)$$

- P^{ME} denotes installed main engine power for the ship.
- $SFOC^{ME}$ denotes the specific fuel consumption (SFOC) for main engines of the ship (kg fuel/kWh).
- LF_x^{Me} denotes the main engine load factor of the ship in the time interval associated with AIS position message x
- Δt_x is the time between AIS position message x and message x – 1 (h).

The load factor is based on the ratio of speed and service speed. The service speed (or design speed) is lower than the maximum speed and hence a sea margin is therefore built into the model. The load factor is not allowed to be more than 1.

$$LF_x^{Me} = \left(\frac{\Delta d_x / \Delta t_x}{SS}\right)^3 \quad (2)$$

- $\Delta d_x$is the great circle distance between AIS position message x-1 and message x (in nm).
- SS denotes the service speed of the ship (in knots).

In this study, only the propulsion power, i.e. $P^{ME} LF_x^{ME}$, was considered and SFOC was hence not considered. The installed main engine power and the service speed of the ship was taken from the IHS Fairplay database.

3. Data description

The main data sources used in this study are the AIS data, ship performance measurement data, the IHS Fairplay database and weather data. The AIS data provide a high-resolution data set with ship position, time and sailing speed for each identified ship, e.g. almost all ships above 300 gross tonnes. The IHS Fairplay database is a ship registry that contain information on particulars, capabilities and vessel types. The weather data is obtained from the National Oceanic and Atmospheric Administration (NOAA) and contains magnitude and direction for ocean waves and winds. The ship performance measurement data is obtained from a DNV GL in-house database comprised of 4209 ships that have reported their performance data at different granularity, from noon reports up to 15-minute intervals.

3.1. Data processing

All the previously mentioned data sources need to be synchronized due to different sampling time and rate. The ship performance measurement data has been selected as being the reference granularity for the sampling, because of its low and regular sampling frequency. The sampling frequency of the ship performance measurement data varies between every 15, 30 or 60 minutes. The weather data has hourly resolution. The frequency of AIS data will vary and is shorter near harbors and shore and when ships are travelling at higher speeds. The synchronization process is shown in Figure 2.

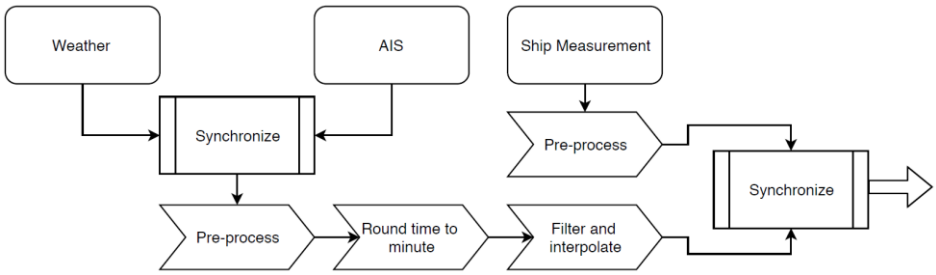

Figure 2. Data synchronization process

First AIS data were synchronized with weather data based on nearest available weather measurement. Then the synchronized weather-AIS data was synchronized with ship measurements based on sampling time. An example of the synchronization result is shown in Figure 3.

Not all the synchronization results are well aligned like in Figure 3. After the synchronization quality check, only four out of ten vessels could be used, and they belong to three different vessel categories. Therefore, for scenario 1 only one single vessel with the most data and satisfactory synchronization result was selected. This ship has 75,000 measurements over a period of 50 months.

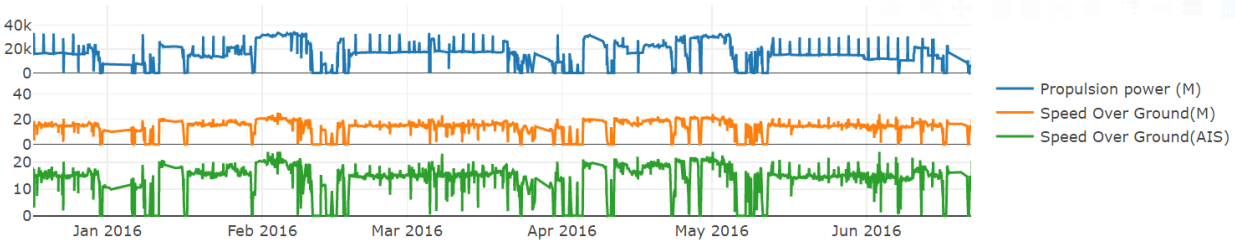

Figure 3. Data synchronization result

Scenario 2 aims to train a model for all container ships and therefore has to take into consideration ship dimensions. For this scenario, a total of 238 container vessels with approximately 400,000 sampling points were included. Because of lack of availability of weather data for these ships, ship dimension, speed over ground and speed through water data were used in addition to the propulsion power. The distribution of the training and test data are shown in Figure 4. 210 of the 238 ships (with 70% of the data based on rows not number of ships) were used as training data, whereas the remaining

28 ships were used as test data. The data are not well balanced, with four vessels contributing to more than half of the data points.

The outliers for the data have been removed based on the vessel information from IHS Fairplay. E.g., the propulsion power is not allowed to be more than the installed main engine power. The direction of the wind and waves has been altered to be relative to the direction of the vessel instead of global coordinates. Normalization has also been applied to the data before the training process to reduce the effect of different variables in different range.

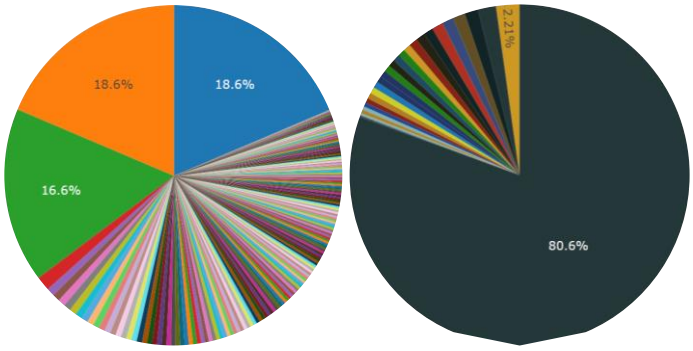

Figure 4. Distribution of the training and test data

It is obvious that the prediction of propulsion power is a time-series problem as the previous speed, power and weather will affect the present power. As such there is a possibility to apply RNN, multi-step MLP or other complex ANN methods for time-series prediction. While, based on the exploration of data for scenario 1, the time gap between measurements are evident from Figure 5 to be highly varying. This also applies to scenario 2 with every 15 minute as maximum resolution.

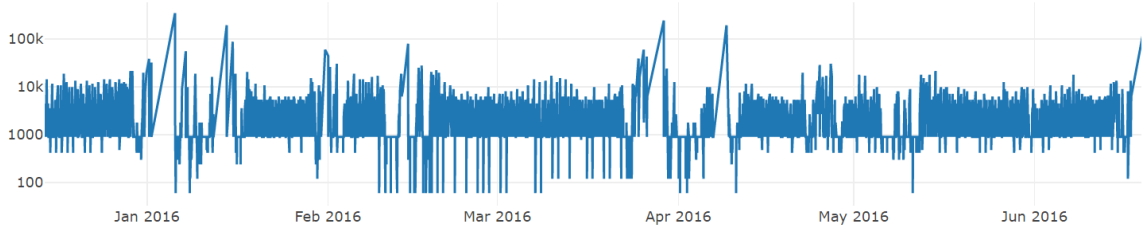

Figure 5. Time gap between measurements in seconds

Based on the exploration of the data, the unbalanced sampling rate disqualify for time-series models. Hence, MLP, also called fully connected neural network, was selected in this study.

3.2. Feature selection

Based on the data available, variables with strong correlation were used to provide a better result. These features have also been discussed and verified with experts to confirm the correlation. The following features (input parameters) were adapted (in Table 1):

Table 1. Features selected for Scenario 1 and 2

Scenario 1			Scenario 2		
Features	Unit	Correlation	Features	Unit	Correlation
Speed over ground	knots	0.89	Speed over ground	knots	0.77
Significant wave height	m	0.39	Speed through water	knots	0.77
Wave direction to vessel	degrees	0.04	Trim	m	-0.19
Wind speed	m/s	0.21	Draft fore	m	0.49
Wind direction to vessel	degrees	0.04	Draft aft	m	0.52
			Design draught	m	0.42

Moulded depth	m	0.49
Moulded breadth	m	0.49
Length between perpendiculars	m	0.52

4. Models training and prediction

In this study, the prediction of propulsion power is a regression problem. Hence the R^2 score, also called coefficient of determination, was selected as the metric to compare the performance of the models. It ranges from 0 to 1, and the higher the R^2 score, the better and more accurate prediction was made. The modelling was carried out in the Scikit-learn 0.19.1 and Keras 2.2.4 library of Python.

4.1. Decision trees and random forest

The Scikit-learn library already provides a good default setting for the parameters to control the learning process which is also called hyperparameters. In this study, selected important parameters were tuned and tested to find the best model. The parameters tuned were the number of trees in the forest, the learning rate, the loss function in the Gradient Boosting Decision Tree (GDBT) algorithm, the fraction of samples to be used for each tree, the maximum number of features to consider when dividing, the maximum depth, the minimum number of samples required for internal node subdivision, the minimum number of samples for each leaf and the maximum number of leaf nodes.

In the hyperparameter tuning process, the models were trained five times to calculate the average R^2 score. In each tuning step, only 1 hyperparameter was tuned to investigate the relationships between hyperparameters and performance. There is no perfect solution for the tuning process, and the combination of hyperparameters which fits the data is the best solution. The combination of hyperparameters adapted shown in Table 2.

Table 2. Scenario 1 and 2 hyperparameters

Model	Hyperparamers:	Hyperparamers:
Gradient Boosting Decision Tree	loss='huber', learning_rate=0.1, n_estimators=1000, subsample=1.0, min_samples_split=2, max_depth=3	loss='ls', learning_rate=0.2, n_estimators=1000, subsample=1.0, min_samples_split=5, max_depth=3
AdaBoost Decision Tree	loss='square'，learning_rate=0.1, n_estimators=1000, splitter='random', min_samples_split=2, max_depth=100	loss='square'，learning_rate=0.1, n_estimators=1000, splitter='random', min_samples_split=3, max_depth=100
Random Forest	n_estimators=1000, min_samples_split=2, min_samples_leaf=5, random_state=None	n_estimators=1000, min_samples_split=5, min_samples_leaf=10, random_state=100

4.2. Support Vector Machine

Normally three kernels are adapted for SVM. It defines how the data was mapped to high dimensional space. In this study, RBF kernel was selected as the kernel function for the SVR model. In addition, linear and polynomial kernel resulted in a very slow training process during the kernel testing process. This is because the training data is difficult to separate by a linear kernel and because a quite complex data structure (huge memory cost) is created by a polynomial kernel [13].

There are two hyperparameters that were defined, whereas the remaining hyperparameters used default values from Scikit-learn. The penalty factor C, which show how much outlier are valued, was set to 1. The gamma value, which is inversely related to the number of support vectors, was set to ‘auto’.

4.3. Neural Network models

As mentioned above, MLP with different structures were adapted for prediction. Models with different numbers of neurons and number of layers were compared. The MLP gives nonlinear relationship between input features and output data. More layers result in more complicated nonlinear transformations. The activation function, kernel initializer and optimizer affect the performance of the ANN. In this study, these factors were not focused on as the parameters had already been tested with a better performance than the other alternatives in one-variable tests. All the kernel initializer settings were set to 'normal'. The activation function settings in the hidden layers were set to 'relu'. The activation function setting in the output layer is 'linear'. The adaptive moment estimation short for 'adam' was selected as optimizer setting.

4.3.1. ANN architecture

If the structure is complex, the training will be extremely slow and difficult. However, the present problem is not as complex as image recognition, which requires complex structure with many layers. In general, MLP does not need to be very deep. One to three hidden layers are generally enough to learn abstract expression from data. As a starting point, the number of neurons in the hidden layer should typically be the average of input and output parameters. The models trained for both scenarios are show in Table 3.

Table 3. MLP models for both scenarios. The model name refers to number of neurons in each layer.

Model	Model Name	Layers	S1 parameters	S2 parameters
1	M_1000	1	7001	12001
2	M_100	1	701	1201
3	M_50	1	351	601
4	M_30	1	211	361
5	M_20	1	141	241
6	M_10	1	71	121
7	M_5	1	36	61
8	M_500_100	2	53201	55701
9	M_100_50	2	5701	6201
10	M_50_30	2	1861	2111
11	M_10_5	2	121	171
12	M_50_20_10	3	1541	1791
13	M_10_8_4	3	189	239
14	M_10_8_6_4	4	235	285

5. Results

5.1. Traditional machine learning result analysis

The prediction of test data for traditional machine learning models is shown in Table 4. Both decision tree related models and random forest model achieved very good predictions for both scenarios, but the SVR model with RBF kernel did not perform well for any combination of hyperparameters. The SVR model even provided negative prediction for propulsion power even though that is not possible. This may be further investigated.

Table 4. Traditional machine learning R^2 score for Scenario 1 and 2

Model name	Scenario 1 R^2 score	Scenario 2 R^2 score
Adaboost Decision Tree	0.759	0.835
Gradient Boosting Decision Tree	0.772	0.847
Random Forest	0.764	0.840
SVR-RBF	0.710	0.766

The performance of the other traditional machine learning models matches the ANN models, see Table 4. However, lots of effort is required to optimize the models to perform well.

5.2. ANN result analysis

Each of the models in Table 3 have been trained five times to calculate the average R^2 score. The results are shown in Figure 6. The name of the model indicates the number of neurons in each layer and hence also the number of layers.

For all ANN model in scenario 1, an increasing number of neurons in each layer resulted in worse performance. Furthermore, models with one hidden layer performs as well or even better than models with two to four layers. Hence, a simple model is preferred.

There is a significant drop for M_500_100, which is caused by overfitting of the data. This model with 55,701 parameters starts to fit for the extreme peak values rather than the general behavior.

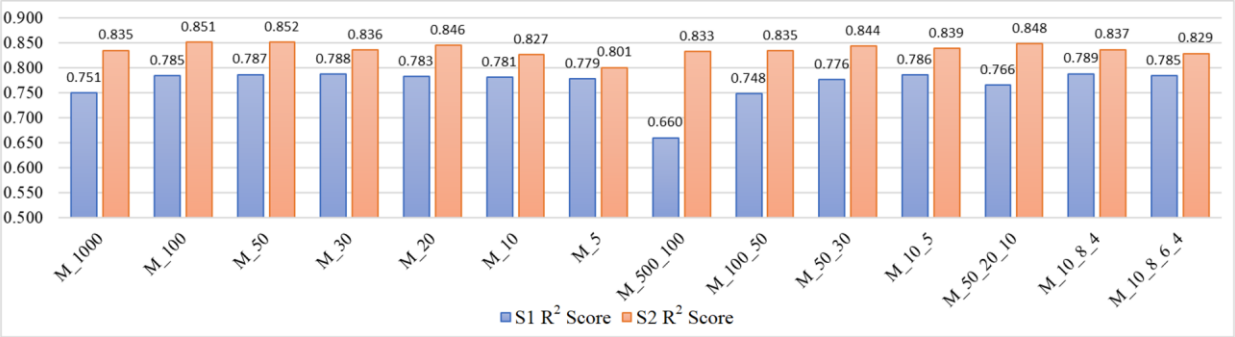

Figure 6. MLP R^2 score for Scenario 1 and 2

Scenario 2 has more features and data. In general, R^2 scores were better than for scenario 1. All the models showed nearly equal performance for different numbers of neurons and layers. However, there are differences, and the best performance was not achieved for the lowest number of neurons. The model with five neurons (M_5) follows the general rule for the number of neurons in a layer, but the best performance was achieved for 50 neurons before the performance slowly dropped when increasing to 1000 neurons. Hence, the general rule can only be used as a starting point before tuning. In general, a simpler model is preferred, and the model with 20 neurons has less than half the parameters of the model with 50 neurons. This 20 neurons model may therefore be considered the best model despite having slightly lower R^2 value.

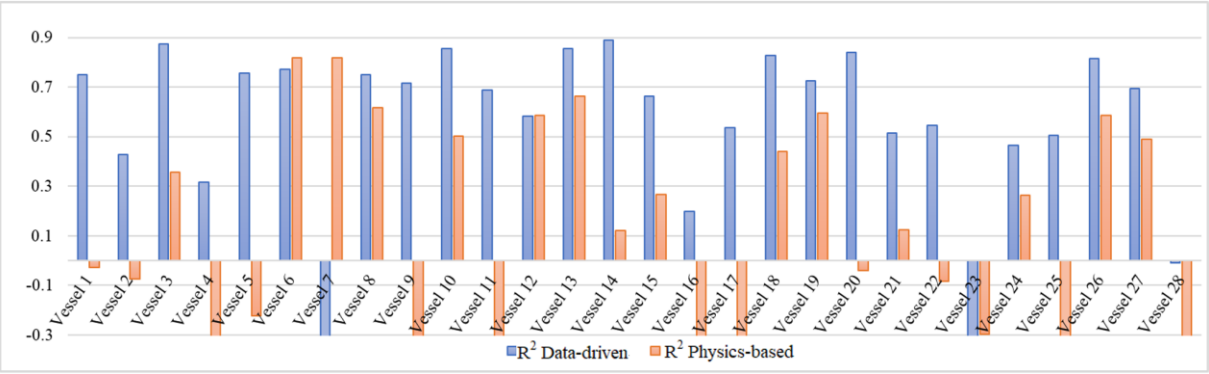

Figure 7. Scenario 2: R^2 score for data-driven and physics-based models

In scenario 2, data for 238 container vessels were included. 28 of them were separated from the training data as test data. Then predictive performance of the physics-based model and data-driven

model (M_20) are shown individually in Figure 7. The overall performance of data-driven model is significantly better than the physics-based model for the vast majority of the vessels (23 out of 28).

The results from three vessels are visualized in Figure 8. For the first vessel (vessel 1), both models match well with the measured propulsion power. The physics-based model has some overshooting problem, which may be caused by the fact that this model strongly relies on the speed, i.e. the speed in the third order, and an incorrect design speed or inaccurate speed reading may amplify the error.

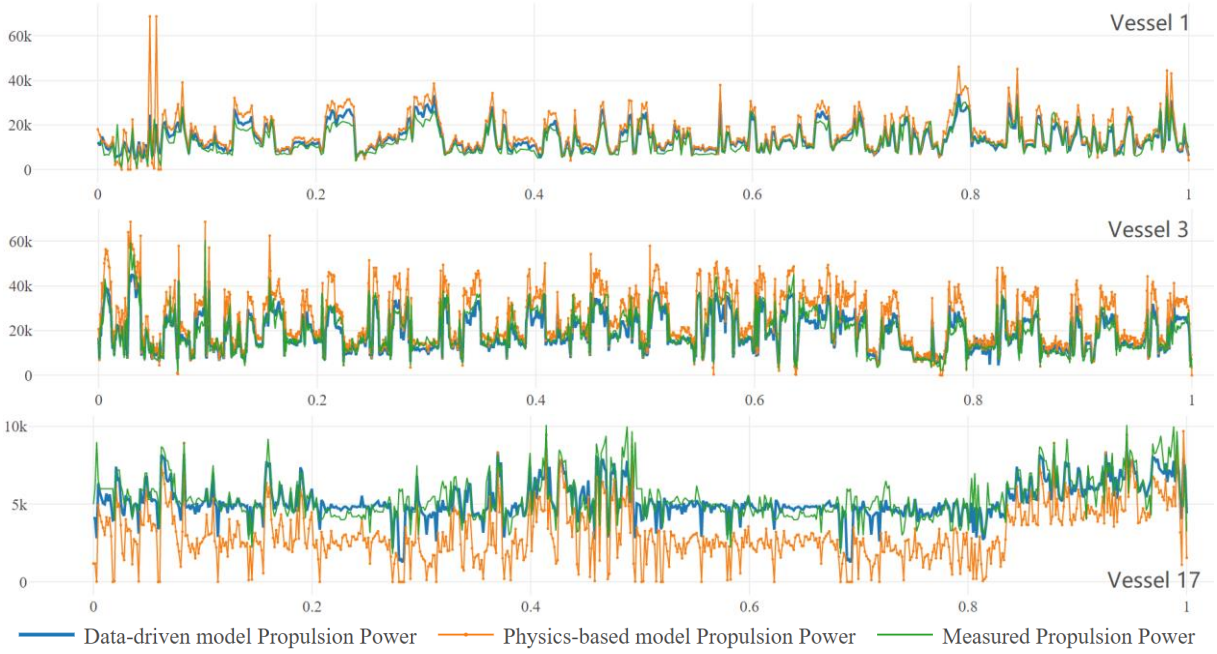

Figure 8. Performance of models with Scenario 2 test data

For the second vessel (vessel 3), both models are generally changing values at the correct time. However, the physics-based model does not capture the magnitude well. This explains the difference in the R^2 between the data-driven model (0.87) and the physics-based model (0.36).

For the third vessel (vessel 17), the physics-based model made predictions much lower than the measurements. This vessel is a smaller container vessel compared to the two vessels mentioned earlier. In scenario 1, because the limitation of data, the model cannot make good prediction for the smaller ships. In scenario 2, more data from different vessels and more dimension related features were adapted. The data-driven model can also perform good prediction on small size vessels. The data-driven model achieved a R^2 score of 0.54 much better than -2.43 from physics-based model.

The difference between these three vessels is that the first two vessels are much bigger than the third vessel with bigger engines which provide more propulsion power. The data-driven model can predict well on the bigger vessels also on the smaller ones which shows different result comparing to scenario 1. This emphasize the importance of data for the data-driven models. If the data covers well on the different sizes of ships and dynamic behaviors, the data-driven model could make accurate prediction on ship operation for ships with different sizes.

5.3. Comparison of ANN and physics-based model

Both for scenario 1 and for all test ships in scenario 2, the fit was clearly better with data-driven models than the physics-based model as shown in Table 5.

The propulsion power calculation in the physics-based model mainly rests on a 3rd order correlation between actual speed from AIS-data and the design speed of a ship. The machine learning methods ability to capture complex relationships from the data was tested by comparing the use of the speed directly in the ANN models with ANN models with speeds in 2nd and 3rd order. The ANN model

architecture was an MLP model with one hidden layer of 20 neurons, and all the results were calculated based on the average of ten trainings and predictions. The results are shown in Table 5.

Table 5. Data intervention result on test data

	Scenario 1	Scenario 2
Physics-based	47.93%	68.59%
Original speed	78.31%	83.83%
Speed by 2nd order	78.69%	83.93%
Speed by 3rd order	78.53%	84.93%

For Scenario 1, there are no significant changes in the R^2 score. For Scenario 2, use of speed in the 3rd order showed a slightly better fit than speed in the first order, but most of the predictive power was achieved already by using speed in the 1st order.

5.4. Ability to predict power for different vessel types

In Scenario 1, only one container ship was used to train the model and the vessel dimension data were not included because they were constant. The model focused on speed, wave and wind, i.e. the dynamic behaviour and influence from the environment. Nevertheless, for another container vessel with similar dimension and another passenger vessel with similar dimensions, for which the model was not trained, the ANN model of scenario 1 still provided better prediction than the physics-based model as shown in Table 6.

Table 6. Prediction results of scenario 1 MLP model

	Physics-based	Trained MLP
1.Target vessel (Container)	48%	78%
2.New container vessel	61%	72%
3.New passenger vessel	54%	82%
4.New general cargo vessel	72%	-58%

For the last ship, the trained model from scenario 1 does not work well. This ship is a general cargo vessel and is much smaller than the vessel of which the training data is based upon. This vessel has consequently a significantly lower power demand. Low power operation is also observed in the training data, but only when the ship is not in transit mode. As a conclusion, and as expected, the simpler scenario 1, without any vessel dimension data, only works on similar sized vessels.

6. Discussion

The performance of physics-based and data-driven models were compared under two scenarios. Scenario 1 focused more on the vessel operation and environment, i.e. contribution from waves and wind on a single vessel. Scenario 2 focused on the general prediction of the container vessels. In both scenarios, data-driven models showed better result than the physics-based models.

As the name indicates, data-driven models strongly rely on the data it has been trained by. If the training data are not representative, the model will not necessarily make good prediction in situations where it has no experience. However, ‘experience’ may be transferred from one vessel to the next. In scenario 1 in this study, the model can only make good predictions on the same or similar sized vessels. For scenario 2, the predictive power is higher for a larger range of container ships.

The distribution of the data also affects the model performance. If the groups have similar amount of data, then the model can make good prediction on all groups. On the contrary, if one ship dominates the amount of data, the predictions may be biased towards this ship, and the predictive power be worse for the remaining ships. The complexity of the data features defines which method to be used. If the problem is not too complex, both traditional machine learning and deep learning may be used. In this study, both methods got equally good performance in both scenarios.

When there is lack of domain knowledge for feature engineering, machine learning provides us with a short cut as it focuses only on the input and output. The traditional machine learning methods need more consideration regarding the tuning of the hyperparameters. In addition, the application environment also defines which method to be used. If the model is running online and is being trained with streamed data, deep learning fits better.

In this study, the performance of MLP models with different structures were compared. The neural network does not need to be complex to provide better performance. In fact, more nodes and layers decreased the performance in many cases. A single hidden layer model is good enough for the regression problems in this study. With regards to estimating how many neurons should be added to the hidden layer, more than the average of input and output performed best. Consequently, the model that can balance both the structure complexity and the performance is a good model.

In the continuation of this work, a combination of scenario 1 and 2 into a scenario 3 will be made. The features will be: the ship dimensions (from the IHS Fairplay database), speed from the AIS data and weather data (i.e. wave, wind and possibly current), which will be calibrated to measured fuel consumption data for the ships. This way the features of the neural network will be same as the physic-based model used in this work, and the results can be compared more directly. In addition to this, a more detailed physical model will be available. Hence, in a scenario 3, neural networks will be compared to a simple physical model and more detailed physical model adjusted for added resistance. The latter will provide an excellent benchmark for the performance of data-driven models compared to physics-based model, i.e. whether data-driven models can outperform the best available physics-based model based on the same input parameters.

Acknowledgements

This work was funded by the VERDE (short for VERification for DEcarbonisation) project, which is funded by the Research Council of Norway by grant #282293. Special thanks to Bingjie Guo in DNV GL for her advice and guidance with the weather data processing and to DNV GL Maritime Advisory for providing the environment data. Acknowledgements to Bjørn Andreas Kristiansen, Master student at NTNU, for laying the ground work of this paper during in the summer project in DNV GL.

References

[1] Chryssakis C, Brinks H, Brunelli A C, Fuglseth T P, Lande M, Laugen L, Longva T, Raeissi B and Tvete H A 2017 Low Carbon Shipping Towards 2050 1-32
[2] Najafabadi M M, Villanustre F, Khoshgoftaar T M, Seliya N, Wald R and Muharemagic E 2015 Deep learning applications and challenges in big data analytics *J. Big Data* **2** 1–21
[3] Leifsson L T, Sævarsdóttir H, Sigurdsson S T and Vésteinsson A 2008 Grey-box modeling of an ocean vessel for operational optimization *Simul. Model. Pract. Theory* **16** 923–932
[4] Parkes A I, Sobey A J and Hudson D A 2018 Physics-based shaft power prediction for large merchant ships using neural networks *Ocean Eng.* **166** 92–104
[5] Petersen J P, Jacobsen D J and Winther O 2012 Statistical modelling for ship propulsion efficiency *J. Mar. Sci. Technol.* **17** 30–39
[6] Bal Beşikçi E, Arslan O, Turan O and Ölçer A I 2016 An artificial neural network based decision support system for energy efficient ship operations *Comput. Oper. Res.* **66** 393–401
[7] Bai S, Kolter J Z and Koltun V 2018 An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling
[8] Pjedsted B and Force P 2009 Modeling of Ship Propulsion Performance *World Marit. Technol. Conf.* 1–10
[9] Shubham Gupta Decision Tree Tutorials & Notes | Machine Learning | HackerEarth
[10] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel V, and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer P, and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos A and and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay E 2011 Scikit-learn: Machine Learning in Python *J. Mach. Learn. Res.* **12** 2825–2830
[11] Indresh Bhattacharyya Support Vector Regression Or SVR – Coinmonks – Medium
[12] Mjelde A, Martinsen K, Eide M and Endresen Ø 2014 Environmental accounting for Arctic shipping – A framework building on ship tracking data from satellites *Mar. Pollut. Bull.* **87** 22–28
[13] Chih-Wei Hsu, Chih-Chung Chang C-J L and Chih-Wei Hsu, Chih-Chung Chang and C-J L 2008 A Practical Guide to Support Vector Classification *BJU Int.* **101** 1396–1400