

Article

# Neural Network Approach for Predicting Ship Speed and Fuel Consumption

Lúcia Moreira \* , Roberto Vettor  and Carlos Guedes Soares 

Centre for Marine Technology and Ocean Engineering (CENTEC), Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal; roberto.vettor@centec.tecnico.ulisboa.pt (R.V.); c.guedes.soares@centec.tecnico.ulisboa.pt (C.G.S.)

\* Correspondence: lucia.moreira@centec.tecnico.ulisboa.pt

**Abstract:** In this paper, simulations of a ship travelling on a given oceanic route were performed by a weather routing system to provide a large realistic navigation data set, which could represent a collection of data obtained on board a ship in operation. This data set was employed to train a neural network computing system in order to predict ship speed and fuel consumption. The model was trained using the Levenberg–Marquardt backpropagation scheme to establish the relation between the ship speed and the respective propulsion configuration for the existing sea conditions, i.e., the output torque of the main engine, the revolutions per minute of the propulsion shaft, the significant wave height, and the peak period of the waves, together with the relative angle of wave encounter. Additional results were obtained by also using the model to train the relationship between the same inputs used to determine the speed of the ship and the fuel consumption. A sensitivity analysis was performed to analyze the artificial neural network capability to forecast the ship speed and fuel oil consumption without information on the status of the engine (the revolutions per minute and torque) using as inputs only the information of the sea state. The results obtained with the neural network model show very good accuracy both in the prediction of the speed of the vessel and the fuel consumption.



**Citation:** Moreira, Lc.; Vettor, R.; Guedes Soares, C. Neural Network Approach for Predicting Ship Speed and Fuel Consumption. *J. Mar. Sci. Eng.* **2021**, *9*, 119. <https://doi.org/10.3390/jmse9020119>

Academic Editor: Decheng Wan  
Received: 29 December 2020  
Accepted: 21 January 2021  
Published: 25 January 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** weather routing; navigation data; artificial neural networks; ship's speed

## 1. Introduction

In the past few years, great attention has been devoted to the efficiency of ship operation in different ocean-going conditions [1,2]. To decrease the greenhouse gas effect, enhancement of the hull shape design [3,4] and the power plant arrangement [5,6] has been conducted in order to reduce ship emissions and fuel consumption. This has been complemented with weather ship routing systems in order to avoid sea states in which ships are forced to increase emissions and fuel consumption [7,8].

There are numerous research approaches on the prediction of ship speed, some addressing the physical mechanisms [9,10], while others deal with statistically based algorithms [11–14]. Normally, ship performance is studied through the speed–power curve of the specified vessel obtained from a series of sea trials. Unfortunately, until ship efficiency is examined, the hub is on fuel consumption, in place of the engine power originated from the speed–power curve. The single curve obtained from sea trials is not enough to study the fuel consumption of the whole range of operating conditions, during the always changing vessel operational schedule. In [15], the authors investigated the problem of fuel consumption prediction and providing the best value for the trim of a vessel in real operation based on data measured onboard. For the fuel consumption prediction, the authors used three different approaches for comparison: white, black, and gray box models. In fact, the influence or trim can also be determined from numerical models to be used at the design stage [16], including the influence of biofouling on ship resistance [17], as well as the uncertainty in predictions [18].

All this generates unreliable estimation conditions [19]. The speed prediction plays a fundamental role in cruise optimization, which is an operation to choose the ideal journey for the vessel crew with regard to augmenting power efficiency and bringing down greenhouse gas emissions in the maritime sector. A correct forecast of ship performance in-operation is a requirement to fulfill these goals [20]. In [21], an innovative platform was introduced to harmonize, through big data technologies, data collected from various sensors onboard. Extreme scale processing techniques were also implemented in order to achieve operational efficiency and performance optimization. This platform was further benchmarked on a series of pilot demonstrations regarding fuel consumption prediction. In [22] an accurate regression model for the fuel consumption of the main engine, using an artificial neural network (ANN), was proposed by big data analysis including data collection, clustering, compression, and expansion. Various numbers of hidden layers and neurons and different types of activation functions were tested in the ANN in order to obtain an accurate regression model, studying their effects on the accuracy and efficiency of the regression analysis. The authors have shown that the proposed regression model using ANN is a more accurate and efficient model to predict the fuel consumption of the main engine than polynomial regression and support vector machines.

The ship speed prediction problem in real sea conditions may be essential and challenging as well [9,13]. Concerning the evaluation of the ship speed in the ocean, it is appropriate to compute added resistance in waves with precision. The added resistance is originated by the radiating waves together with the reflection or diffraction of incident waves. The former mostly arises out of vessel motions and is expertly foreseen with potential theory (strip theory for bow waves), which involves either a far-field technique subjected to the momentum conservation principle, or a near-field technique incorporating the hydrodynamic pressure on the hull surface [23]. However, the forecast of the added resistance generated mostly from a vessel forepart reveals great dispersion of outcomes depending on the adopted approach. There are various techniques available to predict the added resistance in waves, such as potential-based methods [24–27], experimental methods [28,29], and computational fluid dynamics (CFD) approaches [30–34].

The methodology developed and presented here is based on a neural network trained with suitable simulated data acquired through routing software that provides realistic time histories of ship performance at sea [7,8]. These navigation data simulate the data set that would be collected on board [35], and will be adopted to predict the speed of the vessel in different operational conditions. Neural networks have been applied to a range of problems related to ship performance, such as weather forecast [36], wind loading [37], ship maneuvering [38,39], and ship motions and loads [40].

This learning method is very convenient, since it does not need to state the theoretical form of the function, making the calculation procedure expeditious and adequate for being carried out in problem-solving time. Here, the multilayer perceptron (MLP) is adopted, which is among the frequently applied neural net configurations. Considering the system training, the Levenberg–Marquardt backpropagation algorithm is used and finally compared with those obtained with the resilient backpropagation method. The inputs in use to train the ANN concerning the prediction of the ship speed are the output torque of the main engine, the revolutions per minute (RPM) of the propulsion shaft, the significant wave height, and the peak period of the waves, along with the relative angle of wave encounter. The ultimate aim is to apply the approach on board vessels.

The structure of the paper is as follows: Section 2 explains the source of data associated with the ship operation with regard to weather conditions; Section 3 presents the ANN model adopted in this study; Section 4 reports the training procedure; Section 5 considers the results obtained, while in Section 6 the conclusions are established.

## 2. Ship Operational Data

Ship performance at sea can vary appreciably from still water to steering in waves up to the extreme design environments as shown in [7,8]. The principal prerequisites for the

evolution of a well-planned weather routing system are accurate atmospheric condition predictions up to the planned arrival time, as well as the ability to reproduce the vessel's way of behaving whatever conditions it meets in the ocean, determining the motions, achievable speed, combustible consumption, and discharges.

At the design stage, the assessment of ship performance relies on models. These models may be empirical or semi-empirical, numerical, or physical in the case of tests in towing tanks [13]. Concerning the vessel's behavior during ocean operation, model scale ship behavior assessments based on model experiments and theoretical computations may be compared with log book data. However, enough studies have still not been executed through examination of the similarities with on-board monitoring information [1]. Due to economic and time constraints, any of those models can only be verified for a relatively limited number of cases, which are in general different from the large variety of conditions in which a ship can be required to operate in her lifetime.

In order to make up for this limitation, and to provide proper data to validate the proposed ANN model, realistic time histories of ship performance at sea have been simulated by route planning software. The software models the ship behavior at sea, accounting for involuntary speed reductions due to added resistance and engine overloading, and voluntary speed reduction due to undesired ship motion and the exceedance of pre-defined sea-keeping limits [41,42]; it performs a multi-objective optimization through a robust evolutionary algorithm aimed at identifying a set of Pareto-optimal routes balancing voyage duration, fuel consumption, and navigational safety. The first version of the software so far implemented [7,8] takes into consideration the variability of sea-state conditions (significant wave height, peak period, and wave direction) and sailing conditions (ship speed and course) along the route, while assuming constant loading conditions; consequently the ANN developed in this work does not consider displacement and trim as input variables. Thus, the preliminary analysis and testing of the method partially maintains the limitation approach.

The transatlantic route between the English Channel and Miami [43] area of a 24,600 t container-ship has been considered in this paper. Aiming at providing a larger range of variability of the operational conditions, the simulated data include summer and spring voyages, in both directions, westwards and eastwards, and with different mission requirements.

### 3. Artificial Neural Network Model

A neural network is a computer or mathematical model that attempts to imitate either the form or functioning features attributed to living neural systems. ANNs are mostly divided connections of adjustable arbitrary processing elements (PEs); in other words, they are made up of a linked set of artificial neurons and transform the data making use of a connectionist perspective for calculation. However, applied in computer equipment, a PE is an elementary addition of multiplications succeeded by a discontinuance (McCulloch–Pitts model). The links' strengths, named weights, may be modified in an attempt to make the end product of the system correspond to a desired output.

Usually, a neural network is a flexible structure that replicates the intrinsic form constructed upon either outer or inner data, which moves throughout the system during the training stage. Adaptation is the capability to replace the network variables conforming to an order (usually, minimizing an error function). Adaptation allows the network to search for the best performance. A neural network is an arbitrary mathematical statistical modelling implementation. It may be utilized for reproducing complicated interconnections linking fed in and sent out information or to detect patterns in information. Allocated data processing possesses the benefits of accuracy, backup, big distribution of calculations, and collaborative computing.

The MLP is the preponderant applied neural network topology. Lippmann [44] is classified among the most dominant recommended bibliography concerning the numerical abilities of MLPs. Generally, for steady pattern categorization, the MLP with a pair of hidden layers is a general pattern classifier. That is, the discriminant analysis might take

whatever form, as needed by the fed-in information collection. After all, while the weights and the output groups are correctly normalized, the MLP reaches the performance of the highest analytical recipient, which is excellent from a classifying viewpoint [45]. Regarding mapping capabilities, the MLP is thought to be capable of estimating random processes.

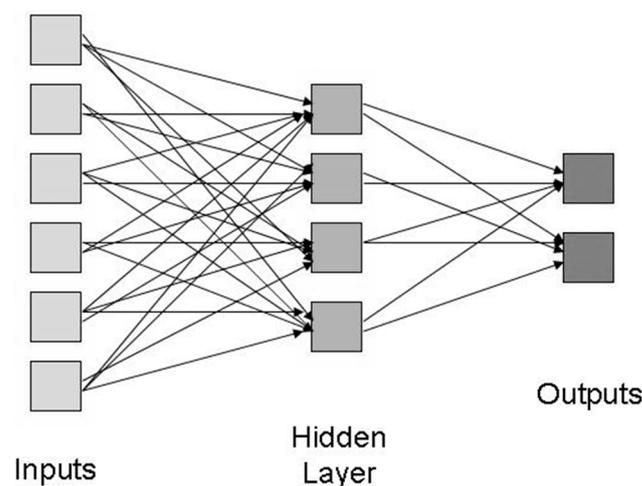
MLPs are usually trained using the backpropagation method and here the Levenberg–Marquardt optimization is applied. The weights and bias are modified conforming the optimization procedure, and the backpropagation technique is utilized for computation of the Jacobian matrix of the performance function with respect to the weights and bias.

Actually, the renovated attentiveness regarding the neural nets has been in part activated due to the existence of backpropagation. The least mean squares (LMS) algorithm presented by Bernard Widrow (1960) may not be applied to hidden PEs, because the wanted signal is unknown in that learning method. The backpropagation algorithm distributes the errors along the system and permits adjustment of the hidden PEs.

Two of the main features of an MLP are its arbitrary PEs, which possess a discontinuance that should be smooth (the logistic function and the hyperbolic tangent are preponderant and universally adopted); and its great connectivity (in other words, a component that belongs to a specified layer supplies every component of the following layer). The MLP is trained using error-correction learning, which signifies that the network required response needs to be found. In pattern identification that is usually the event, because the fed-in information is specified (in other words, which information is part of which test is discovered).

Here a unique-hidden-layer MLP structure is adopted for the time-domain relation connecting outputs and inputs. A common arrangement illustration is shown in Figure 1. A unique bias neuron  $b$  will be summed to every one of the input  $p_j$  and hidden layers that will be added to the weighted inputs  $w_j$  to constitute the net input  $n$ , which can be expressed by

$$n = \sum_{j=1}^n w_j p_j + b \tag{1}$$



**Figure 1.** An illustration of a general elementary feedforward network.

The quantity of the hidden neurons regulates the robustness of the network design. This quantity was attentively selected in the subsequence of several training runs performed for confirmation through a sensitivity analysis of the results.

Here two different structures of MLP systems (feedforward ANNs) were developed in MatLab and trained employing the Levenberg–Marquardt method, and then compared with the resilient backpropagation optimization algorithm. The first MLP consists of three layers of nodes: an input layer, a hidden layer, and an output layer; the second one has an

input layer, three hidden layers, and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function, in this case a log sigmoid function. In the first part of the study, the training procedure used five input nodes and one output node as indicated before. For this first set of results it used only the first MLP structure (one hidden layer) and a formulation of two hidden neurons was chosen in the subsequent study of the system convergence and generalization ability. The training was performed only using the Levenberg–Marquardt method and the adopted input variables were:

- H<sub>s</sub>—significant wave height;
- T<sub>p</sub>—peak period of wave;
- Torque—main engine output;
- RPM—revolutions of the shaft;
- Alpha—relative angle of wave encounter.

The relative angle of wave encounter was reduced to the range <0,180> degrees due to the assumed symmetry of the wave impact on the resultant ship speed regardless of the wave approaching from port side or starboard side.

#### 4. Neural Network Training

A log-sigmoid activation function is employed to all neurons belonging to the hidden layer, producing a system with the capability to produce smooth choices. The function is described as follows

$$f(x_i) = \frac{1}{1 + e^{-x_i}} \tag{2}$$

The error correction learning performs as follows: deriving out of the network response at PE *i* at iteration *n*, *y<sub>i</sub>(n)*, and the wanted response *d<sub>i</sub>(n)* for a stated fed-in sample an instantaneous error *e<sub>i</sub>(n)* can be established through

$$e_i(n) = d_i(n) - y_i(n) \tag{3}$$

Backpropagation calculates the sensitivity of a cost function with reference to every weight in the system, and refreshes all weights in proportion to the sensitivity [46]. The benefit of the method is that it may be applied with local data and needs only small products per weight, which is extremely effective. The disadvantage is that since this is a gradient descent method, it exclusively makes use of the local data so it may lay in relative minima. Furthermore, the method is naturally noisy because a deficient approximation of the gradient is adopted, producing slow convergence.

At the same time as backpropagation, a steepest-descent method, the Levenberg–Marquardt algorithm, is carried out, which originated from the Newton–Raphson method that was outlined for minimizing functions that are sums of squares of nonlinear functions [47], of the form

$$E = \frac{1}{2} \sum k(e_k)^2 = \frac{1}{2} \|e\|^2 \tag{4}$$

where *e<sub>k</sub>* is the error in the *k<sup>th</sup>* pattern and *e* is the array of the components *e<sub>k</sub>*. As long as the divergence between the prior weight array and the actual one is small, the error array may be approached to 1st-order through a Taylor series:

$$e(j + 1) = e(j) + \frac{\partial e_k}{\partial w_i} (w(j + 1) - w(j)) \tag{5}$$

Consequently, the error function might be given by

$$E = \frac{1}{2} \|e(j) + \frac{\partial e_k}{\partial w_i} (w(j + 1) - w(j))\|^2 \tag{6}$$

Through minimization of the error function with reference to the actual weight array:

$$w(j + 1) = w(j) - (J^T J)^{-1} J^T e(j) \tag{7}$$

where  $(J)_{ki} = \frac{\partial e_k}{\partial w_i}$  is the Jacobian matrix.

The Hessian for the sum-of-square error function is stated by

$$(H)_{ij} = \frac{\partial^2 E}{\partial w_i \partial w_j} = \sum \left\{ \left( \frac{\partial e_k}{\partial w_i} \right) \left( \frac{\partial e_k}{\partial w_j} \right) + e_k \frac{\partial^2 e_k}{\partial w_i \partial w_j} \right\} \tag{8}$$

Ignoring the 2nd term in (8), the Hessian may be rewritten in the following way:

$$H = J^T J \tag{9}$$

The refreshing of the weights requires the inverse Hessian. The Hessian is moderately simple to calculate, because it is formulated on 1st-order derivatives with respect to the network weights, which are effortlessly adjusted through backpropagation. Despite the fact that the refreshing procedure may be employed iteratively for the minimization of the error function, this can result in a big step size that can cancel the linear estimation where the method is established. In the Levenberg–Marquardt technique, the error function is minimized, while the step size is preserved as small in spite of making sure of the linear estimation viability. This is achieved through adoption of a reformulated error function as follows:

$$E = \frac{1}{2} \|e(j) + \frac{\partial e_k}{\partial w_i} (w(j + 1) - w(j))\|^2 + \lambda \|w(j + 1) - w(j)\|^2 \tag{10}$$

where  $\lambda$  is a variable controlling the step size. Minimizing the modified error with respect to  $w(j + 1)$ :

$$w(j + 1) = w(j) - (J^T J + \lambda I)^{-1} J^T e(j) \tag{11}$$

When the scalar  $\lambda$  is zero, Equation (11) expresses precisely Newton’s method, employing the approximate Hessian matrix. When  $\lambda$  is large, the formulation changes into gradient descent with a small step size. Newton’s method is expeditious and more precise near an error minimum, so the objective is to move in the direction of Newton’s method expeditiously. Consequently,  $\lambda$  is reduced after all favorable steps (performance function reduction) and is augmented only when a preliminary step will increment the performance function. Thus, the performance function is decreased every time at every algorithm iteration.

The sigmoid transfer functions typically used by the multilayer networks in the hidden layers are often called “squashing” functions, because they compress an infinite input range into a finite output range. Sigmoid functions are characterized by the fact that their slopes must approach zero as the input gets large. This causes a problem when the steepest descent is used to train a multilayer network with sigmoid functions, because the gradient can have a very small magnitude and, therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values.

The purpose of the resilient backpropagation training algorithm is to eliminate these harmful effects of the magnitudes of the partial derivatives. Only the sign of the derivative can determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. The size of the weight change is determined by a separate update value. The update value for each weight and bias is increased by a factor whenever the derivative of the performance function with respect to that weight has the same sign for two successive iterations. The update value is decreased by a factor whenever the derivative with respect to that weight changes sign from the previous iteration. If the derivative is zero, the update value remains the same. Whenever the weights are oscillating, the weight

change is reduced. If the weight continues to change in the same direction for several iterations, the magnitude of the weight change increases. A complete description of the resilient backpropagation algorithm is given in [48]. This learning algorithm is used for comparison of the results obtained with the Levenberg–Marquardt method.

## 5. Results of Speed Prediction

The simulated data obtained with the weather routing code utilized to verify the neural net approach were divided into four sets, representing four voyages in the North Atlantic: westward and eastward, in April and July 2001, respectively.

In each simulation run the values of the ship speed, the output torque of the main engine, the RPM of the propulsion shaft, the significant wave height, the peak period of the waves, and the relative angle of wave encounter were estimated at consecutive control-points along the route located at a distance of about 30 nautical miles, roughly corresponding to one hour and a half at the considered ship speeds; however, the sampling rate was not kept constant. To better understand the characteristics of each voyage the respective maximum, minimum, mean, and standard deviation values of the significant wave height, peak period, and ship's speed are listed in Table 1.

**Table 1.** Maximum, minimum, and average values of  $H_s$  and  $T_p$  in each voyage.

Set		$H_s$ (m)	$T_p$ (s)	Speed (knots)
01042001_east	Min	0.8	5.3	17.5
	Max	4.9	14.2	20
	Mean	2.4	10.4	19.1
	StDev	0.8	1.5	0.5
01042001_west	Min	0.8	4.9	15.2
	Max	5	13.5	20
	Mean	2.7	10.5	18.7
	StDev	1	1.6	0.9
01072001_east	Min	0.7	4.5	18.1
	Max	1.9	9.9	20
	Mean	1.2	8.7	19.2
	StDev	0.2	0.9	0.5
01072001_west	Min	1	5.8	17.3
	Max	3.3	11.2	20
	Mean	1.7	9	19.1
	StDev	0.5	0.9	0.6

When training multiple-layer networks, the common procedure is to initially split the information into three subsets:

1. The training set, which is utilized both to calculate the gradient and to refresh the network weights and biases;
2. The error on the validation set is observed throughout the training procedure. The validation error usually drops in the course of the beginning of the procedure, as well as the training set error. Nevertheless, when the model starts to over-fit the information, the error on this set generally increases. The system weights and biases are saved when the validation set error is at a minimum;
3. The last set is the test set. This set error is not employed throughout the training procedure, but it is utilized to measure the differences between distinct models. It is also convenient to outline the test-set error throughout the procedure. If the error on

this set reaches a minimum at a substantially dissimilar iteration number than the validation set error, this may denote a deficient split of the information set.

Here, the function employed to execute the division of the data is automatically obtained when the model is trained, and is utilized to split the information into training, validation, and testing subsets. Afterwards, the data is randomly divided into the three subsets. The fractions of information that are placed in the training, testing, and validation sets are 0.7, 0.15, and 0.15, respectively. The sets have different numbers of training points varying between 12,754 and 15,618 in total for each set.

The correlation analysis is performed using the Pearson correlation coefficient  $r$ , which is calculated to measure the linear dependence between the model and desired outputs. According to the definition, the Pearson correlation coefficient between an output  $x$  and a desired output  $d$  is stated by

$$r = \frac{\sum_i (x_i - \bar{x})(d_i - \bar{d})}{N \sqrt{\frac{\sum_i (d_i - \bar{d})^2}{N}} \sqrt{\frac{\sum_i (x_i - \bar{x})^2}{N}}} \tag{12}$$

The  $r$  values acquired for the evaluation of the vessel speed and fuel oil consumption (FOC) over the training, test, and validation data sets are recorded in Tables 2 and 3, respectively.

**Table 2.**  $r$  results of the neural network training for the speed estimation.

Set	Training	Test	Validation
01042001_east	0.99998	0.99998	0.99998
01042001_west	0.99999	0.99999	0.99999
01072001_east	0.99998	0.99998	0.99998
01072001_west	0.99999	0.99999	0.99999

**Table 3.**  $r$  results of the neural network training for the fuel oil consumption (FOC) estimation.

Set	Training	Test	Validation
01042001_east	0.9997	0.99971	0.99971
01042001_west	0.99979	0.99981	0.9998
01072001_east	0.99963	0.99962	0.99962
01072001_west	0.99961	0.99959	0.99961

It can be seen that very good agreement is achieved for each one of the tested sets utilizing only two hidden neurons. Figures 2 and 3 show the respective regression plots, illustrating the fitness between the outputs and targets for the ship speed estimation.

The labels of the y-axes of Figures 2 and 3 indicate equations between the predicted value and the target value, with output as the dependent variable and target as the independent variable. These equations were employed to express how well the ANN performed. The coefficient of the target reveals the correlation between the output and the targets, consequently for a satisfactory performance of the MLP it should be as close to unity as possible. The second term, which is a constant, is the error or the residue that should be summed to the scaled target to force it as close as possible to the predicted output; preferably it would be zero or as small as possible.

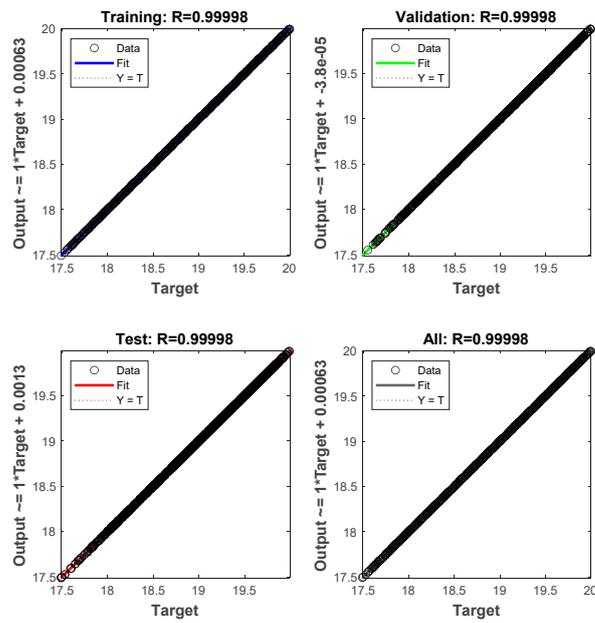


Figure 2. Regression plot of ship speed estimation for 01042001\_east voyage.

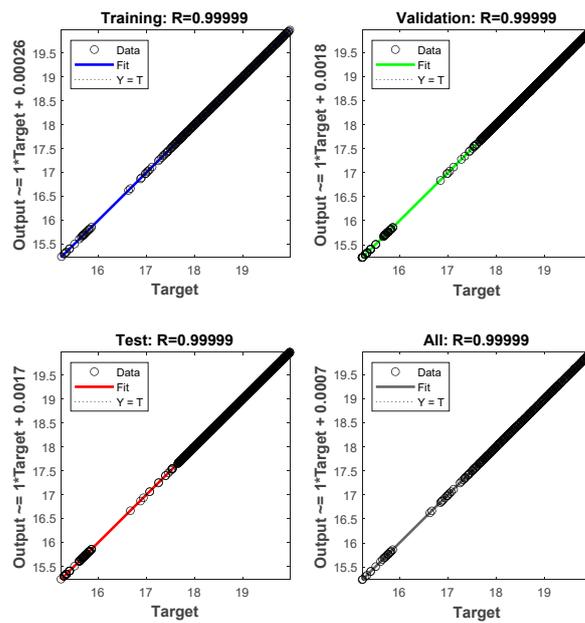


Figure 3. Regression plot of ship speed estimation for 01042001\_west voyage.

The results obtained in the regression plots for 01072001\_east and 01072001\_west voyages for the ship speed estimation are similar to the ones presented in Figures 2 and 3, as well as the regression plots for the FOC estimation, and for this reason these figures are omitted here.

Figures 4 and 5 show comparisons between the neural net prediction and the results derived from the simulations. A total of 1000 data points are exhibited for each data set. The plots show the very good correspondence between the neural net approximations and the results achieved throughout simulations.

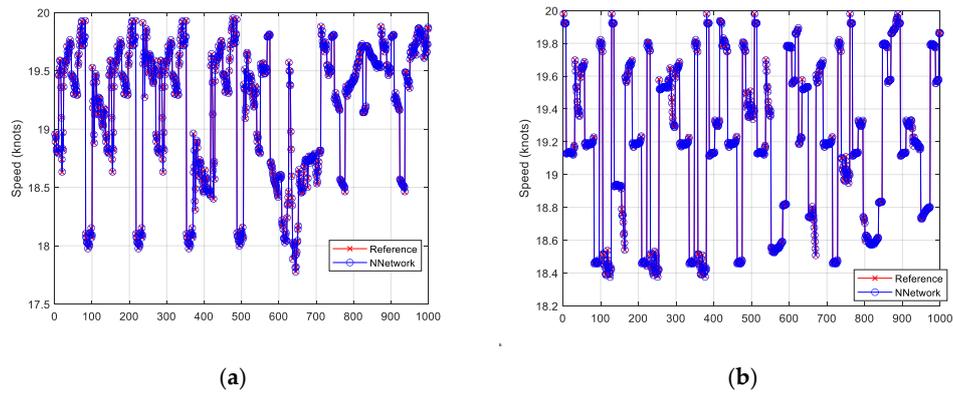


Figure 4. Ship speed estimation for 01042001\_east voyage (a) and 01072001\_east voyage (b).

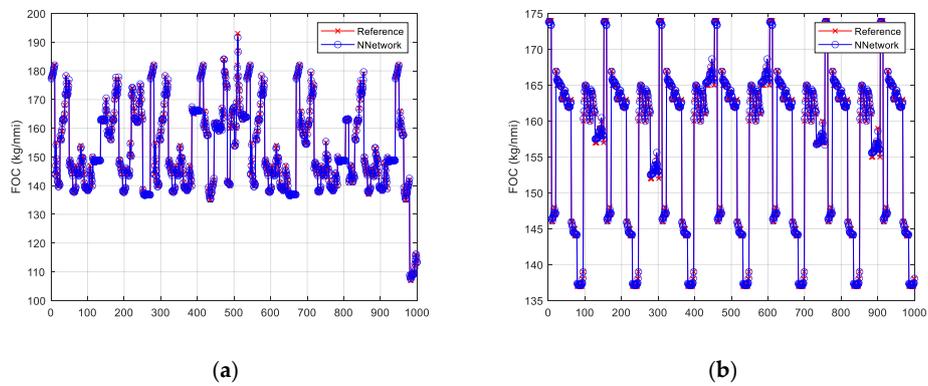


Figure 5. Ship FOC estimation for 01042001\_west voyage (a) and for 01072001\_west voyage (b).

Tables 4 and 5 present the results of the sensitivity analysis performed to analyze the ANN prediction capability to predict the ship speed and FOC, respectively, without information of the engine state (RPM and torque), using as inputs only the sea conditions. This analysis was made for two different neural network structures: using just one hidden layer vs. three hidden layers.

Table 4. *r* results of the neural network training for the speed estimation without engine data (revolutions per minute (RPM) and torque).

Set	# Neurons	Training	Test	Validation	# Neurons	Training	Test	Validation
		1 Hidden Layer			3 Hidden Layers			
01042001_east	500	0.83305	0.79923	0.79979	(10,8,5)	0.73607	0.71825	0.72408
	1500	0.87335	0.79837	0.797	(30,24,15)	0.8711	0.81634	0.81815
	2500	0.89448	0.72459	0.70962	(50,40,25)	0.87359	0.82242	0.82525
	3500	0.90464	0.70748	0.76544	(70,56,35)	0.89292	0.82786	0.82972
01042001_west	500	0.89963	0.867	0.86298	(10,8,5)	0.86514	0.84379	0.82868
	1500	0.93114	0.81858	0.80992	(30,24,15)	0.92734	0.89429	0.90309
	2500	0.94916	0.74497	0.71868	(50,40,25)	0.92925	0.91167	0.91289
	3500	0.96477	0.68316	0.75009	(70,56,35)	0.95202	0.9107	0.93457
01072001_east	500	0.75559	0.64495	0.70445	(10,8,5)	0.72377	0.68608	0.72475
	1500	0.7737	0.67893	0.62169	(30,24,15)	0.79231	0.76462	0.74681
	2500	0.78536	0.48162	0.66675	(50,40,25)	0.81213	0.77395	0.78152
	3500	0.81119	0.47694	0.53975	(70,56,35)	0.84178	0.73745	0.79112
01072001_west	500	0.94481	0.93012	0.93948	(10,8,5)	0.89636	0.90164	0.88516
	1500	0.94767	0.91549	0.88025	(30,24,15)	0.93673	0.93222	0.92256
	2500	0.95184	0.69684	0.84094	(50,40,25)	0.94539	0.91793	0.93351
	3500	0.95679	0.83816	0.80804	(70,56,35)	0.94611	0.94309	0.93069

**Table 5.** *r* results of the neural network training for the FOC estimation without engine data (RPM and torque).

Set	# Neurons	Training	Test	Validation	# Neurons	Training	Test	Validation
		1 Hidden Layer			3 Hidden Layers			
01042001_east	500	0.86868	0.81374	0.80524	(10,8,5)	0.72752	0.70458	0.71658
	1500	0.90968	0.65906	0.7396	(30,24,15)	0.86467	0.84008	0.82613
	2500	0.92912	0.69925	0.63298	(50,40,25)	0.90323	0.831	0.87054
	3500	0.93531	0.68073	0.52335	(70,56,35)	0.91192	0.8493	0.86787
01042001_west	500	0.8921	0.84335	0.83737	(10,8,5)	0.81616	0.79053	0.80964
	1500	0.9186	0.67158	0.67754	(30,24,15)	0.91636	0.88548	0.87724
	2500	0.9341	0.65872	0.53965	(50,40,25)	0.93256	0.88269	0.89747
	3500	0.94808	0.47342	0.49402	(70,56,35)	0.93965	0.91198	0.89295
01072001_east	500	0.76349	0.74112	0.72334	(10,8,5)	0.73169	0.71298	0.71492
	1500	0.79443	0.70283	0.59887	(30,24,15)	0.78155	0.7394	0.75511
	2500	0.81774	0.51816	0.59301	(50,40,25)	0.81687	0.77622	0.77101
	3500	0.81933	0.44761	0.5555	(70,56,35)	0.82196	0.78277	0.79427
01072001_west	500	0.93414	0.82817	0.89365	(10,8,5)	0.87304	0.86794	0.8789
	1500	0.94678	0.8893	0.79715	(30,24,15)	0.93989	0.9094	0.93275
	2500	0.9524	0.69815	0.80717	(50,40,25)	0.94133	0.92397	0.93407
	3500	0.95553	0.52913	0.69541	(70,56,35)	0.94481	0.9311	0.91889

From the obtained results presented in Tables 4 and 5 it can be seen that it is possible to predict the ship speed and the FOC with satisfactory accuracy using as input data only the information about the sea conditions and disregarding the input of the RPM and torque from the engine. A loss of accuracy can be noted in the voyages where the bandwidths of  $H_s$  and  $T_p$  are smaller (lower standard deviation). It is also noted that the model with just one hidden layer is less efficient for more neurons in the test and validation datasets, while the opposite is reported for the training dataset. With the neural network structure with three hidden layers, this problem is solved and the efficiency for the test and validation datasets are almost the same as the one obtained for the training set.

Table 6 presents the results of the sensitivity analysis performed to analyze the ANN prediction capability to predict the ship speed using the resilient backpropagation learning algorithm instead of the Levenberg–Marquardt algorithm, again without information of the engine state (RPM and torque), using as inputs just the sea conditions. This analysis was made for two different neural network structures: using just one hidden layer vs. three hidden layers, as presented in Tables 4 and 5.

**Table 6.** *r* results of the neural network training for the speed estimation without engine data (RPM and torque) using the resilient backpropagation learning algorithm.

Set	# Neurons	Training	Test	Validation	# Neurons	Training	Test	Validation
		1 Hidden Layer			3 Hidden Layers			
01042001_east	500	0.847751	0.80008	0.78189	(10,8,5)	0.74209	0.70515	0.72117
	1500	0.87451	0.8062	0.77896	(30,24,15)	0.84231	0.8144	0.80016
	2500	0.8808	0.74577	0.76869	(50,40,25)	0.88351	0.77954	0.80395
	3500	0.88936	0.74783	0.70206	(70,56,35)	0.8866	0.8161	0.82104
01042001_west	500	0.90911	0.85626	0.87917	(10,8,5)	0.88662	0.87458	0.87339
	1500	0.93746	0.8745	0.8623	(30,24,15)	0.93107	0.89077	0.90269
	2500	0.95700	0.76202	0.81459	(50,40,25)	0.93183	0.9013	0.91107
	3500	0.96479	0.68641	0.59637	(70,56,35)	0.93974	0.90997	0.89098
01072001_east	500	0.75737	0.72569	0.70828	(10,8,5)	0.74846	0.73983	0.74871
	1500	0.77205	0.67415	0.70032	(30,24,15)	0.79078	0.7329	0.74000
	2500	0.80292	0.63342	0.6793	(50,40,25)	0.79492	0.74307	0.77971
	3500	0.80849	0.54946	0.44425	(70,56,35)	0.80905	0.76011	0.76813
01072001_west	500	0.94447	0.92248	0.9266	(10,8,5)	0.91051	0.9216	0.89416
	1500	0.94771	0.88309	0.90539	(30,24,15)	0.93932	0.92596	0.93173
	2500	0.9515	0.84724	0.88683	(50,40,25)	0.94703	0.93701	0.93741
	3500	0.95317	0.74362	0.88727	(70,56,35)	0.94752	0.92143	0.91997

The results obtained with the Levenberg method were compared with another training method, in this case the resilient backpropagation learning algorithm, in order to analyze its significance. From the obtained results presented in Table 6 and comparing with the ones presented in Table 4, it can be seen that the magnitudes of the results are similar as well as the behavior of the models when the inner structure is changed.

## 6. Conclusions

A procedure based on artificial neural networks has been implemented to provide estimations of ship speed and FOC from the output torque of the main engine, the RPM of the propulsion shaft, the significant wave height, the peak period of the waves, and the relative angle of wave encounter measurements. The obtained neural network system is suitable for producing precise approximations of both variables, indicating that it is feasible to achieve acceptable results with an ANN trained using only two hidden neurons. Additionally, it was demonstrated that it is possible to predict the ship speed and the FOC using as input data only information about the sea conditions. The main purpose of the presented approach is for it to be integrated into a hull monitoring system. The ANN can be used as a redundant system in case of failure of a sensor providing the foreseen information for the controllers, and can help in anticipating some ship speed deviation from predefined target conditions, reducing action delays optimizing the ship operation.

The information utilized to adjust and confirm the validity of the neural net system was acquired through simulations of ship operations executed with a routing program providing a large set of practical navigation data to copy the data set collected on board. Thus the neural network can be used as an alternative to the more complex system used in [7,8], or any other weather routing system [49,50]. It is also a good option to be used in decision support systems onboard, like the ones described in [35,51].

**Author Contributions:** The concept of the problem was developed by C.G.S. and L.M. The simulation data used as the input to the ANN were generated by R.V. The ANN analysis was performed by L.M., and the writing of the original draft manuscript was done by L.M. and revised by C.G.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Portuguese Foundation for Science and Technology (Fundação para a Ciência e Tecnologia—FCT) under grants on the ROUTING research project (MARTERA-1/ROUTING/3/2018) in the ERA-NET COFUND MarTERA-1 program (2018–2021). The paper contributes to the Strategic Research Plan of the Centre for Marine Technology and Ocean Engineering (CENTEC), which is financed by the Portuguese Foundation for Science and Technology (Fundação para a Ciência e Tecnologia—FCT) under contract UIDB/UIDP/00134/2020.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

ANN	Artificial neural network
CFD	Computational fluid dynamics
FOC	Fuel oil consumption
LMS	Least mean squares
MLP	Multilayer perceptron
PE	Processing element
RPM	Revolutions per minute

## References

1. Tsujimoto, M.; Orihara, H. Performance prediction of full-scale ship and analysis by means of onboard monitoring (Part 1 Ship performance prediction in actual seas). *J. Mar. Sci. Technol.* **2018**, *24*, 16–33. [[CrossRef](#)]
2. Papanikolaou, A.; Zaraphonitis, G.; Bitner-Gregersen, E.; Shigunov, V.; El Moctar, O.; Guedes Soares, C.; Reddy, D.N.; Sprenger, F. Energy efficient safe ship operation (SHOPERA). *Transp. Res. Procedia* **2016**, *14*, 820–829. [[CrossRef](#)]
3. Cheng, X.; Feng, B.; Liu, Z.; Chang, H. Hull surface modification for ship resistance performance optimization based on Delaunay triangulation. *Ocean Eng.* **2018**, *153*, 333–344. [[CrossRef](#)]

4. Hou, Y.H. Hull form uncertainty optimization design for minimum EEOI with influence of different speed perturbation types. *Ocean Eng.* **2017**, *140*, 66–72. [[CrossRef](#)]
5. Tadros, M.; Ventura, M.; Guedes Soares, C. Optimization procedure to minimize fuel consumption of a four-stroke marine turbocharged diesel engine. *Energy* **2019**, *168*, 897–908. [[CrossRef](#)]
6. Tadros, M.; Ventura, M.; Guedes Soares, C. Optimization of the performance of marine diesel engines to minimize the formation of SOx emissions. *J. Mar. Sci. Appl.* **2020**, *19*, 473–484. [[CrossRef](#)]
7. Vettor, R.; Guedes Soares, C. Development of a ship weather routing system. *Ocean Eng.* **2016**, *123*, 1–14. [[CrossRef](#)]
8. Vettor, R.; Guedes Soares, C. Characterisation of the expected wave conditions in the main European coastal traffic routes. *Ocean Eng.* **2017**, *140*, 244–257. [[CrossRef](#)]
9. Guedes Soares, C.; Fonseca, N.; Ramos, J. Prediction of Voyage Duration with Weather Constraints. In Proceedings of the International Conference on Motions and Manoeuvrability, Royal Institute of Naval Architects, London, UK, 18–19 February 1998; pp. 1–13.
10. Prpić-Oršić, J.; Vettor, R.; Faltinsen, O.M.; Guedes Soares, C. The influence of route choice and operating conditions on fuel consumption and CO<sub>2</sub> emission of ships. *J. Mar. Sci. Technol.* **2016**, *21*, 1–24. [[CrossRef](#)]
11. Mao, W.; Rychlik, I.; Wallin, J.; Storhaug, G. Statistical models for the speed prediction of a container ship. *Ocean Eng.* **2016**, *126*, 152–162. [[CrossRef](#)]
12. Gan, S.; Liang, S.; Li, K.; Deng, J.; Cheng, T. Long-term ship speed prediction for intelligent traffic signalling. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 1–10. [[CrossRef](#)]
13. Krata, P.; Vettor, R.; Guedes Soares, C. Bayesian approach to ship speed prediction based on operational data. In *Developments in the Collision and Grounding of Ships and Offshore Structures*; Guedes Soares, C., Ed.; Taylor & Francis Group: London, UK, 2020; pp. 384–390.
14. Perera, L.P.; Mo, B. Ship speed power performance under relative wind profiles in relation to sensor fault detection. *J. Ocean Eng. Sci.* **2018**, *3*, 355–366. [[CrossRef](#)]
15. Oneto, L.; Baldi, F.; Anguita, D.; Coraddu, A. Vessels fuel consumption forecast and trim optimisation: A data analytics perspective. *Ocean Eng.* **2016**, *130*, 351–370. [[CrossRef](#)]
16. Islam, H.; Guedes Soares, C. Effect of trim on container ship resistance at different ship speeds and drafts. *Ocean Eng.* **2019**, *183*, 106–115. [[CrossRef](#)]
17. Garcia, S.; Trueba, A.; Boullosa-Falces, D.; Islam, H.; Guedes Soares, C. Predicting ship frictional resistance due to biofouling using Reynolds-averaged Navier-Stokes simulations. *Appl. Ocean Res.* **2020**, *101*, 102203. [[CrossRef](#)]
18. Islam, H.; Guedes Soares, C. Uncertainty analysis in ship resistance prediction using OpenFOAM. *Ocean Eng.* **2019**, *191*, 105805. [[CrossRef](#)]
19. Kee, K.K.; Yew, S.L.B. Prediction of ship fuel consumption and speed curve by using statistical method. In Proceedings of the 7th International Conference on computer Science and Computational Mathematics ICCSCM2018, Pulau Langkawi, Malaysia, 5–6 July 2018.
20. Lu, R.; Turan, O.; Boulougouris, E.; Banks, C.; Incecik, A. A semi-empirical ship operational performance prediction model for voyage optimization towards energy efficient shipping. *Ocean Eng.* **2015**, *110*, 18–28. [[CrossRef](#)]
21. Spandonidis, C.; Panagiotis, T.; Giordamli, C. Combined multi-layered big data and responsible AI techniques for enhanced decision support in shipping. In Proceedings of the International Conference on Decision Aid Sciences and Application (DASA 2020), Sakheer, Bahrain, 8–9 November 2020.
22. Jeon, M.; Noh, Y.; Shin, Y.; Lim, O.-K.; Lee, I.; Cho, D. Prediction of ship fuel consumption by using an artificial neural network. *J. Mech. Sci. Technol.* **2018**, *32*, 5785–5796. [[CrossRef](#)]
23. Kim, Y.; Kim, K.; Kim, J.; Kim, Y.; Park, I.; Jang, Y. Analysis of added resistance and seakeeping responses in head sea conditions for low-speed full ships using URANS approach. *Int. J. Nav. Archit. Ocean Eng.* **2017**, *9*, 641–654. [[CrossRef](#)]
24. Liu, S.; Papanikolaou, A.; Zaraphonitis, G. Prediction of added resistance of ships in waves. *Ocean Eng.* **2011**, *38*, 641–650. [[CrossRef](#)]
25. Matulja, D.; Sportelli, M.; Guedes Soares, C.; Prpic-Oršic, J. Estimation of added resistance of a ship in regular waves. *Brodogradnja* **2011**, *62*, 259–272.
26. Soding, H.; Shigunov, V. Added resistance of ships in waves. *Ship Technol. Res.* **2015**, *62*, 2–13. [[CrossRef](#)]
27. Vitali, N.; Prpić-Oršić, J.; Guedes Soares, C. Uncertainties related to the estimation of added resistance of a ship in waves. In *Maritime Technology and Engineering 3*; Guedes Soares, C., Santos, T.A., Eds.; Taylor & Francis Group: London, UK, 2016; pp. 391–399.
28. Park, D.; Lee, J.; Jung, Y.; Lee, J.; Kim, Y. Experimental and numerical studies on added resistance of ship in oblique sea conditions. *Ocean Eng.* **2019**, *186*, 106070. [[CrossRef](#)]
29. Wang, W.; Wu, T.; Zhao, D.; Guo, C.; Luo, W.; Pang, Y. Experimental-numerical analysis of added resistance to container ships under presence of wind-wave loads. *PLoS ONE* **2019**, *14*, e0221453. [[CrossRef](#)] [[PubMed](#)]
30. Hu, C.; Mikami, T.; Yamamoto, K. Prediction of added resistance in short waves by CFD simulation. In Proceedings of the 29th International Workshop on Water Waves and Floating Bodies, Osaka, Japan, 30 March–2 April 2014.
31. Kim, M.; Hizir, O.; Turan, O.; Incecik, A. Numerical studies on added resistance and motions of KVLCC2 in head seas for various ship speeds. *Ocean Eng.* **2017**, *140*, 466–476. [[CrossRef](#)]

32. Kim, M.; Hizir, O.; Turan, O.; Day, S.; Incecik, A. Estimation of added resistance and ship speed loss in a seaway. *Ocean Eng.* **2017**, *141*, 465–476. [[CrossRef](#)]
33. Ozdemir, Y.H.; Barlas, B. Numerical study of ship motions and added resistance in regular incident waves of KVLCC2 model. *Int. J. Nav. Archit. Ocean Eng.* **2017**, *9*, 149–159. [[CrossRef](#)]
34. Islam, H.; Guedes Soares, C.; Liu, J.; Wang, X. Propulsion power prediction for an inland container vessel in open and restricted channel from model and full-scale simulations. *Ocean Eng.* **2021**. [[CrossRef](#)]
35. Perera, L.P.; Rodrigues, J.M.; Pascoal, R.; Guedes Soares, C. Development of an onboard decision support system for ship navigation under rough weather conditions. In *Sustainable Maritime Transportation and Exploitation of Sea Resources*; Rizzuto, E., Guedes Soares, C., Eds.; Taylor & Francis Group: London, UK, 2012; pp. 837–844.
36. Campos, R.M.; Krasnopolsky, V.; Alves, J.H.; Penny, S.G. Improving NCEP's global-scale wave ensemble averages using neural networks. *Ocean Model.* **2020**, *149*, 101617. [[CrossRef](#)]
37. Haddara, M.; Guedes Soares, C. Wind Loads on Marine Structures. *Mar. Struct.* **1999**, *12*, 199–209. [[CrossRef](#)]
38. Moreira, L.; Guedes Soares, C. Dynamic Model of Manoeuvrability using Recursive Neural Networks. *Ocean Eng.* **2003**, *30*, 1669–1697. [[CrossRef](#)]
39. Wang, X.; Guedes Soares, C. Direct Adaptive Neural Network Control for Ship Manoeuvring Modelling Group Model-Based Uncertain Nonlinear Systems in Non-affine Pure-Feedback Form. *IEEE Access* **2020**, *8*, 3272–3284. [[CrossRef](#)]
40. Moreira, L.; Guedes Soares, C. Neural Network Model for Estimation of Hull Bending Moment and Shear Force on Ships in Waves. *Ocean Eng.* **2020**, *206*, 107347. [[CrossRef](#)]
41. Fonseca, N.; Guedes Soares, C. Sensitivity of the Expected Ships Availability to Different Seakeeping Criteria. In Proceedings of the 21st International Conference on Offshore Mechanics and Arctic Engineering (OMAE'02), Oslo, Norway, 23–28 June 2002. [[CrossRef](#)]
42. Guedes Soares, C. Effect of Heavy Weather Manoeuvring on the Wave Induced Vertical Bending Moments in Ship Structures. *J. Ship Res.* **1990**, *34*, 60–68. [[CrossRef](#)]
43. Vettor, R.; Guedes Soares, C. Detection and analysis of the main routes of voluntary observing ships in the North Atlantic. *J. Navig.* **2015**, *68*, 397–410. [[CrossRef](#)]
44. Lippmann, R.P. An introduction to computing with neural nets. *IEEE Trans. ASSP Mag.* **1987**, *4*, 4–22. [[CrossRef](#)]
45. Makhoul, J. Pattern recognition properties of neural networks. In Proceedings of the 1991 IEEE Workshop on Neural Networks for Signal Processing, Princeton, NJ, USA, 29 September–2 October 1991. [[CrossRef](#)]
46. Hush, D.R.; Horne, B.G. Progress in supervised neural networks. *IEEE Signal Process. Mag.* **1993**, *10*, 8–39. [[CrossRef](#)]
47. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin, Germany, 2006.
48. Riedmiller, M.; Braun, H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In Proceedings of the IEEE International Conference on Neural Networks (ICNN'93), San Francisco, CA, USA, 28 March–1 April 1993. [[CrossRef](#)]
49. Simonsen, M.H.; Larsson, E.; Mao, W.G.; Ringsberg, J.W. State-of-the-art within ship weather routing. In Proceedings of the 34th ASME International Conference on Ocean, Offshore and Arctic Engineering (OMAE2015), ASME paper UNSP V003T02A053, St John's, NL, Canada, 31 May–5 June 2005.
50. Perera, L.P.; Guedes Soares, C. Weather Routing and Safe Ship Handling in the Future of Shipping. *Ocean Eng.* **2017**, *130*, 684–695. [[CrossRef](#)]
51. Rodrigues, J.M.; Perera, L.P.; Guedes Soares, C. *Decision Support System for the Safe Operation of Fishing Vessels in Waves*; Maritime Engineering and Technology; Guedes Soares, C., Garbatov, Y., Sutulo, S., Santos, T.A., Eds.; Taylor & Francis Group: London, UK, 2012; pp. 153–161.