

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

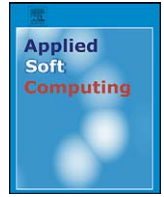
In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Predictions of oil/chemical tanker main design parameters using computational intelligence techniques

S. Ekinci^{a,b,c,d}, U.B. Celebi^{a,b,c,d}, M. Bal^{a,b,c,d}, M.F. Amasyali^{a,b,c,d,*}, U.K. Boyaci^{a,b,c,d}

^a Yildiz Technical University, Faculty of Naval Architecture and Maritime, Naval Architecture and Marine Engineering Department, Istanbul, Turkey

^b Yildiz Technical University, Faculty of Chemistry and Metallurgical, Mathematical Engineering Department, Istanbul, Turkey

^c Yildiz Technical University, Faculty of Electric and Electronics, Computer Engineering Department, Istanbul, Turkey

^d National Research Institute of Electronics and Cryptology, TUBITAK, UEKAE, Gebze, Kocaeli, Turkey

ARTICLE INFO

Article history:

Received 26 March 2009

Received in revised form 26 July 2010

Accepted 16 August 2010

Available online 12 October 2010

Keywords:

Computational intelligence

Machine learning

Neural networks

Naval engineering

Ship design main parameters

ABSTRACT

Ship design and construction are complicated and expensive processes. In the pre-design stage, before the construction according to some special rules, determination of the main ship parameters is very important. In this study, instead of traditional methods, the oil/chemical tanker main design parameters are estimated by 18 computational intelligence methods. Therefore, all the data of 114 tankers in operation are used in the experiments in order to estimate a parameter from the remaining ones. Main result of this article is to show that, except for the speed parameter, the main parameters of tankers can be estimated sufficiently well for pre-design stage without having to apply conventional but arduous ship modeling experiments.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Hydrodynamic model tests are the most conventional methods in Naval Architecture to obtain ship main dimensions and main engine power, coupled to resistance-propulsion characteristics, accurately. During model test, geometric model of ship, scaled down to required a size, is towed along towing tank at specified speeds to measure data for ship resistance, propulsion, engine power calculations. But, model production takes time and is costly due to labor and material needs. In this case, to obtain required data for ships with different dimensions in each time a model ship should be produced and repeated tests should be conducted. Nonetheless, in this study prediction of main design parameters and main engine power for ships outside of present data cluster using Computational Intelligent Techniques, instead of experimental method, is presented.

The most important stage before the production of a chemical tanker construction is the pre-design stage. In the pre-design, with respect to the capacity of goods, the main parameters of the tankers, namely, length, height, draught, breadth, main engine power and

speed must be determined as correctly, rapidly and inexpensively as possible. The parameters in focus are calculated using the parameters above mentioned as main design parameters. An error in the calculation of the main parameters consequently results in situations difficult to return or costly to compensate. In order to estimate the main engine power, conventional methods imposed model pool experimental data and/or highly professional software results are used. The convenient methods, especially the experiments, can take enormous times. To overcome, recent machine learning techniques have been investigated.

Of parameters utilized at initial design stage, length between perpendiculars (L_{BP}), breadth (B), draught (T) and deadweight ton (DWT) have influence on stability, strength, capacity, construction and operational costs, moreover, ship speed (V) and main engine power (P) have influence on techno-economic aspects such as, hydrodynamic characteristics, resistance/propulsion, seakeeping and they all affect performance of ships. Thus, at the initial design stage, it is crucial that, these parameters should be determined accurately. For instance, hull weight increases with increasing (L_{BP}), resulting an increase in total construction cost. Or, increase in B value causes higher resistance and power. Or, decrease in T results in longitudinal strength challenges. Considering these facts, predicting and determining the relation of coupled parameters which have importance at initial design stage and the ship speed and engine power are more of an issue.

The study of Alkan et al. [1], in which two fishing vessel data were used in several algorithms in order to calculate the ship geom-

* Corresponding author.

E-mail addresses: ekinci@yildiz.edu.tr (S. Ekinci), ucelebi@yildiz.edu.tr (U.B. Celebi), mertbal@yildiz.edu.tr (M. Bal), mfatih@ce.yildiz.edu.tr (M.F. Amasyali), ukasif@uekae.tubitak.gov.tr (U.K. Boyaci).

etry and stability, can be regarded as a pioneer for a realistic neural network in this field.

Amasyali et al. [2] and Bal et al. [3] made main power engine estimations with an acceptable rate of success, in their studies. These previous works have given light to the idea that the pre-design parameters for the chemical tankers, despite their enormous difference in size with motorboats can be estimated by computational intelligence methods as well. In this study various algorithms have been used, not just a few, but all of the parameters have been estimated well.

Recent studies related to artificial neural network and machine learning methods find practice in naval architecture and ocean engineering fields: In particular, identification of ship motion characteristics using ship hydrodynamics, fault optimization of propeller and shaft systems, intelligent ship autopilot design [4–10], prediction of immersed ship form underwater acoustic noise and ship classification [11–15] and prediction, simulation and fault analysis of ship propulsion powers [16–18].

In this study, different Computational Intelligence Techniques are applied for determining principal parameters of current oil/chemical tankers which have lengths of 53–182 m.

This article is organized as follows. In the following chapter, definition of ship parameters are given. The third chapter defines some properties of the data set. In the fourth chapter, artificial intelligence methods deployed in this study are explained briefly. In the last two chapters, the experimental results and a summary of ideas that may be deduced from this study are given, respectively.

2. Tanker' main dimensions and used tanker data sets

Tanker' profile and front view are given in Fig. 1. Main dimensions and related tanker parameters are defined as follows.

- (BL) **Baseline**: The horizontal line parallel to the design waterline (DWL), which cuts the midship section at the lowest point of the ship. The vertical heights are usually measured from the baseline.
- (AP) **After Perpendicular**: The vertical line at the point of intersection of the LWL and the centerline of the rudderstock.
- (FP) **Forward Perpendicular**: The vertical line at the point of intersection of the LWL and the forward end of the immersed part of the ship's hull.
- (LWL) **Load Water Line**: The water line at which the ship will float when loaded to its designed draught.
- ($L=L_{BP}$) **Length Between Perpendiculars**: The distance measured parallel to the base at the level of the load waterline from the after perpendicular to the forward perpendicular.
- (T) **Draught**: The vertical distance from the waterline at any point on the hull to the bottom of the ship.
- (B) **Breadth**: The distance from the inside of plating on one side to a similar point on the other side measured at the broadest part of the ship.
- (V) **Ship Speed (Knot)**: The distance in miles taken in a hour.
- (DWT) **DeadWeight Ton**: Total amount of weight that a ship can carry (cargo, fuel, lubricating oil, fresh water, stores, passengers and baggage, crew and their effects).
- (P) **Main Engine Power(kW)**: This is measured at the flywheel of an engine.

In this study, 114 real oil/chemical tanker data (obtained from www.veristar.com web site and www.gisbir.com – Turkish shipbuilders association web site) are used. Each tanker's main parameters L_{BP} , DWT, B, T, V, P are estimated by using computational methods. The parameters' cross-correlation are given in Table 1. The scatter plot of the data used is given by Fig. 2. Table 1 and Fig. 2 are examined together, and the highest correlation observed is between B and L_{BP} . Moreover, the average highest and lowest

Table 1

Cross-correlation of pre-design parameters.

	L_{BP}	B	T	V	P
DWT	0.9265	0.9547	0.9318	0.5018	0.8872
L_{BP}		0.9724	0.9615	0.6808	0.9627
B			0.9670	0.6156	0.9367
T				0.6510	0.9373
V					0.7418

correlation between the remaining main parameters belong to T and V, respectively.

3. Computational Intelligence Techniques

In this study, several regression algorithms from different software packages are used. Table 2 illustrates list of methods used and related software packages. WEKA is available at www.cs.waikato.ac.nz/ml/weka. Regression Toolbox is available at www.control.hut.fi/Hyotyniemi/publications/report125/RegrToolbox, PLStoolbox is available at www.eigenvector.com. Matlab Neural Network Toolbox version 7.0 is used. The notation used in the methods are:

The regression approximation addresses the problem of estimating a function $y=f(x)$ based on a given data set $G = \{(x_i, d_i)\}_{i=1}^N$ where $x_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ is input vector, d_i is the desired value, y_i estimated value, f is the estimation function, w is the weight vector, N is the number of observations and m is the number of features.

3.1. Least Median Square (LMS)

LMS algorithm uses the estimation of the gradient vector from the given data set. Successive corrections to the weight vector in the direction of the negative of the gradient vector are made iteratively until reaching the minimum median square error.

3.2. Simple Linear Regression (SLR)

It is the process of fitting straight lines (models) between each attribute and output. In Eq. (1), the values of w and w_0 are estimated by the method of least squares.

$$y = wx + w_0 \quad (1)$$

The model having lowest squared error is selected as the final model among each parameter model.

3.3. Linear Regression (LR)

The final model is linear regression of a subsample of the attributes. The subsample is selected by iteratively removing the one with the smallest standardized coefficient until no improvement is observed in the estimate of the error given by the Akaike information criterion [22].

$$y = w_0 + w_i x_i + \varepsilon \quad (2)$$

The AIC value, with the assumption of eventual normally distributed errors, is calculated as follows,

$$AIC = 2k + N \ln \left(\frac{RSS}{N} \right) \quad (3)$$

where k is the number of parameters in the model, RSS is the Residual of Sum of the Squares.

3.4. Multi-layer Perceptron with Adaptive Learning Rate (GDA)

In this technique, backpropagation is often used. The backpropagation used in this article is derived from a Least Mean

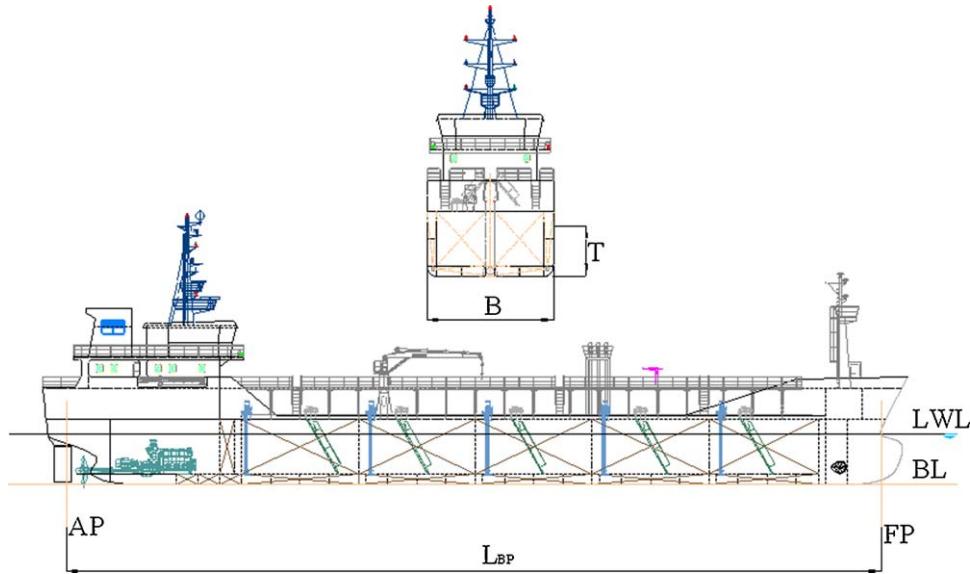


Fig. 1. An oil/chemical tanker's profile and front view.

Squares Approach described in [23]. The implementation algorithm is derived from [24]. If the bias terms are considered as an input always equal to 1 and fed through a weight, then the output equation of the basic element of the network matrix is given by

$$\text{Output}_j = S \left(\sum_{i=0}^m w_{ij} x_i \right) \quad (4)$$

where w_{ij} is the weight connecting input i to neuron j and S is a sigmoid non-linearity function with the formula,

$$S(a) = \frac{1}{1 + \exp^{-a}} \quad (5)$$

First, the error for the output layer nodes is computed by

$$E_j = (d_j - a_j) a_j (1 - a_j) \quad (6)$$

where E_j is error for node j of the output layer, d_j is target activation for node j of the output layer and a_j is actual activation for node i of the output layer. Then, successively, the error values for all hidden

layer nodes are computed by using

$$E_i = a_i (1 - a_i) \sum_j E_j w_{ij} \quad (7)$$

where E_i is error for node i in a hidden layer, E_j is error for node j in the layer above, w_{ij} is weight for the connection between node i in the hidden layer and node j in the previous layer and a_i is activation of node i in the hidden layer.

At the end of the backward propagation phase, nodes of the network (except the input layer nodes) will have error values. The error value of a node is used to compute new weights for the connections which lead to the node. The weight change is given by

$$w_{ij} = w_{ij} + \Delta w_{ij} \quad (8)$$

where w_{ij} is the weight of the connection between node i in the previous layer and node j in the output layer or in a hidden layer and Δw_{ij} is the weight change that is given by

$$\Delta w_{ij} = n_t a_j E_i \quad (9)$$

where t is the epoch number. The current learning rate n_t is not allowed to be greater in magnitude than z times the previous learn-

Table 2
Used regression algorithms

Method name	Abbreviation	Software
Least Median Squared Linear Regression	LMS	WEKA
Linear Regression	LR	
Pace Regression	PR	
Radial Basis Functions	RBF	
Support Vector Regression	SMO	
Simple Linear Regression	SLR	
M5 Model Trees Regression	M5P	
M5 Model Trees Rules	M5R	
Neural Network (gradient descent with adaptive learning rate)	GDA	Matlab Neural Network Toolbox
Neural Network (gradient descent with momentum)	GDM	
Neural Network (Levenberg–Marquardt optimization)	LM	
Generalized regression neural network	GRNN	
Multiple Linear Regression	MLR	Regression Toolbox
Orthogonal Least Regression	OLS	
Ridge Regression	RR	
Total Least Squares	TLS	
Partial Least Squares	PLS	PLS Toolbox
Principal Components Regression	PCR	

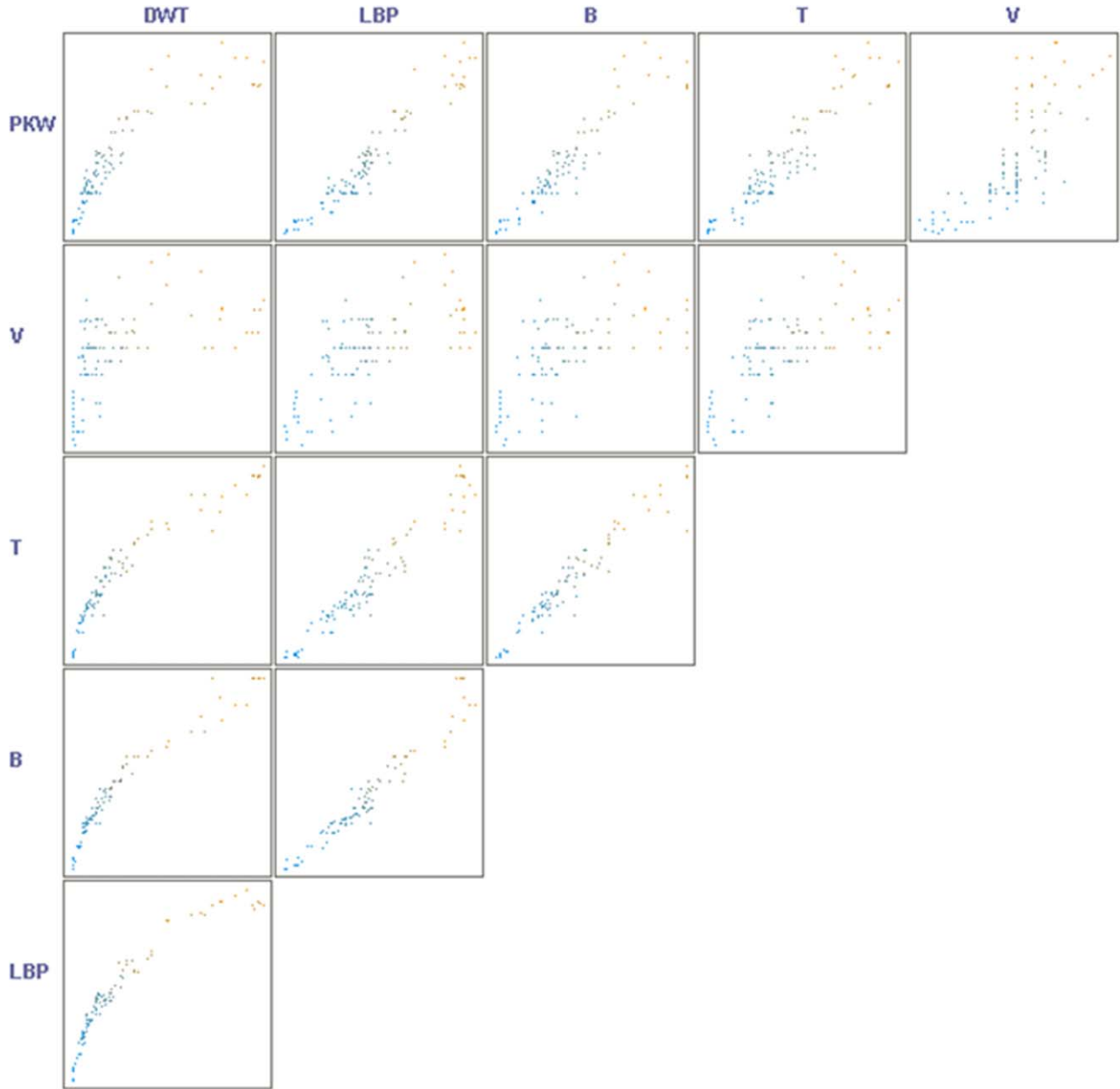


Fig. 2. Main design parameters' scatter plot.

ing rate n_{t-1} . The selection of initial n_t and z is crucial step. In our experiments, constant learning rate is used ($n_t = 0.01$ - Matlab Neural Network Toolbox's default value).

3.5. Multi-layer Perceptron with Momentum (GDM)

The only difference between this method and GDA method is the calculation of the weight change. Momentum simply adds a fraction (m) of the previous weight update to the current one. The weight change in GDM method is given by

$$\Delta w_{ij}(t) = -n_t E(w_{ij}(t)) + m \Delta w_{ij}(t-1) \quad (10)$$

3.6. Radial Basis Function (RBF) Neural Network

For a one-node output layer, the global input–output relationship of an RBF Neural Network can be expressed as a linear

combination of K basis function as follows,

$$f(\vec{x}) = \sum_{k=1}^K w_k \phi_k(\vec{x}) \quad (11)$$

where $\vec{x} = [x_1, x_2, \dots, x_m]^T$ is the m -dimensional input vector, w_k are the weighting coefficients of the linear combination, and $\phi_k(\vec{x})$ represents the response of the k th neuron of the hidden layer. Typically, the basis function $\phi_k(\vec{x})$ are assumed to be Gaussian shaped with scale factor σ_k ; their values decrease monotonically with the distance between the input vector \vec{x} and the center of each function $\vec{c}_k = [c_{k1}, c_{k2}, \dots, c_{km}]^T$ [25].

$$\phi_k(\vec{x}) = \exp\left(-\frac{\|\vec{x} - \vec{c}_k\|^2}{\sigma_k^2}\right) \quad (12)$$

3.7. Orthogonal Least Squares (OLS)

OLS is an efficient implementation of the forward stepwise model selection technique for linear regression model [33]. Before selecting a subset model, all the regressors of the initial model are studied in order to determine how each regressor will contribute to the modeling of the desired system. The regressor selected to be included in the subset models is the one that will contribute most significantly with the regressor already selected to model the desired system. The selection process is repeated until the required number of regressor is found. In order to be able to select a RBF network model, an initial network with a large number of nodes must be formed. Each of the node parameters are then pre-set to fix the hidden layer's response. By fixing the hidden layer's response, the RBF network is reduced to a linear regression model. The generalized RBF network with a single output node is described by

$$y = w_0 + \sum_{j=1}^K w_j \phi(\tilde{x}|\lambda_j) \quad (13)$$

where $y \in \mathbb{R}$ is the response of the network to the input vector $x \in \mathbb{R}^m$, $\{w_j\}_{j=0}^K$ are the scalar weights, $\phi(\cdot)$ is a radial basis function and λ_j is the set of parameters for the j th RBF node. Gaussian non-linearity is used to realize $\phi(\cdot)$, i.e.

$$\phi(\tilde{x}|\lambda_j) = \exp\left(-\frac{|\tilde{x} - \tilde{c}_j|^T \tilde{D}_j |\tilde{x} - \tilde{c}_j|}{\sigma_j}\right) \quad (14)$$

where $\lambda_j = \{\sigma_j, \tilde{c}_j, \tilde{D}_j\}$. The variable σ_j determines the width of the Gaussian response, $\tilde{c}_j \in \mathbb{R}^m$ is a vector known as the center of the RBF node, and $\tilde{D}_j \in \mathbb{R}^{m \times m}$ is a diagonal matrix. In practice, the parameters σ_j and \tilde{D}_j are pre-set to some initial values and centers \tilde{c}_j are chosen from the data points $\{x_i\}_{i=1}^N$.

3.8. Pace Regression (PR)

PR is an improvement over OLS in models having a large number of attributes many of which are redundant. Pace regression evaluates the effect of each variable and uses a clustering analysis to improve the statistical basis for estimating their contribution to the overall regression.

3.9. Multiple Linear Regression (MLR)

Multiple Linear Regression (MLR) is one of the statistical tools used for discovering relationship between variables. It is used to find the linear model that best predicts the dependent variable from the independent variables [29].

If the engineering data are created by numerous variables, the analytical methods become very complicated. If a function is a linear function containing two or more variables, the MLR method is an excellent approach to fit engineering data. For example, let y be the linear fitting function of the m number of variables x_1, x_2, \dots, x_m , i.e., $y(x_1, x_2, \dots, x_m) = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_m x_m$ where w_0, w_1, \dots, w_m are constant parameters numbered at $m+1$. If the number of engineering data is N , then the sum of the square of error is $S_r = \sum_{i=1}^N (y_i - w_0 - w_1 x_{1,i} - w_2 x_{2,i} - \dots - w_m x_{m,i})^2$. Derive the derivatives of the above equation with respect to each parameter, and let

$$\frac{\partial S_r}{\partial w_0} = \frac{\partial S_r}{\partial w_1} = \dots = \frac{\partial S_r}{\partial w_m} \quad (15)$$

then a matrix equation can be derived as follows:

$$\begin{bmatrix} n & \sum x_{1,i} & \sum x_{2,i} & \dots & \sum x_{m,i} \\ \sum x_{1,i} & \sum x_{1,i}^2 & \sum x_{2,i} x_{1,i} & \dots & \sum x_{1,i} x_{m,i} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum x_{m,i} & \sum x_{m,i} x_{1,i} & \sum x_{m,i} x_{2,i} & \dots & \sum x_{m,i}^2 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_{1,i} y_i \\ \vdots \\ \sum x_{m,i} y_i \end{bmatrix} \quad (16)$$

The elements a_0, a_1, \dots, a_m are the equation parameters estimated by fitting engineering data. The estimated Standard Deviation (SD) of MLR is $\sigma_R = \sqrt{S_r / (N - m - 1)}$. Therefore, the sample number N must be at least larger than $m+1$. Otherwise, the above equation will be invalid if the denominator of the square root non-positive.

3.10. Levenberg–Marquardt (LM)

Levenberg–Marquardt method is one of the best known curve-fitting algorithms which works better than the Gauss–Newton method even if the starting point is very far off the desired minimum. To better illustrate the advantage of LM method over Newton method, consider a multilayer feedforward neural network.

The Newton method calculates the weight function recurrently as follows,

$$w_{n+1} = \bar{w}_n - [H\psi(\bar{w}_n)]^{-1} \nabla \psi(\bar{w}_n) \quad (17)$$

where $\psi: \mathbb{R}^K \rightarrow \mathbb{R}$ is the corresponding error function given by,

$$\psi(\bar{w}) = \frac{1}{2} \bar{e}(\bar{w})^T \bar{e}(\bar{w}) \quad (18)$$

In the LM method, instead of the exact calculation of Hessian matrix which is too costly, a positive-definite approximation is calculated. The approximation formula is given by $H\psi(\bar{w}) \approx \bar{J}(\bar{w})^T \bar{J}(\bar{w}) + \mu \bar{I}$ where

$$\bar{J}(\bar{w}) = \begin{bmatrix} \frac{\partial e_{11}(\bar{w})}{\partial w_1} & \frac{\partial e_{11}(\bar{w})}{\partial w_2} & \dots & \frac{\partial e_{11}(\bar{w})}{\partial w_K} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{L1}(\bar{w})}{\partial w_1} & \frac{\partial e_{L1}(\bar{w})}{\partial w_2} & \dots & \frac{\partial e_{L1}(\bar{w})}{\partial w_K} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{1Q}(\bar{w})}{\partial w_1} & \frac{\partial e_{1Q}(\bar{w})}{\partial w_2} & \dots & \frac{\partial e_{1Q}(\bar{w})}{\partial w_K} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{LQ}(\bar{w})}{\partial w_1} & \frac{\partial e_{LQ}(\bar{w})}{\partial w_2} & \dots & \frac{\partial e_{LQ}(\bar{w})}{\partial w_K} \end{bmatrix} \quad (19)$$

is the Jacobian matrix, $\mu > 0$, \bar{I} is an identity matrix of size $K \times K$.

The LM method weight recurrence is then calculated as in [26] by

$$w_{n+1} = \bar{w}_n - [\bar{J}(\bar{w}_n)^T \bar{J}(\bar{w}_n) + \mu_n \bar{I}]^{-1} \nabla \psi(\bar{w}_n) \quad (20)$$

The only drawback of LM method compared to Gaussian method, its slowness in nice conditions, comes from the fact that at each iteration step, the inverse of $\bar{J}(\bar{w}_n)^T \bar{J}(\bar{w}_n) + \mu_n \bar{I}$ has to be calculated.

3.11. General Regression Neural Network (GRNN)

Specht [28] proposed a simple but with a high estimation cost method GRNN which can solve any smooth approximation given enough data. GRNN can be regarded as a normalized RBF network where the hidden layer is usually expressed by a multivariate Gaussian function ϕ with an appropriate mean and an auto variance

matrix as follows:

$$\phi_k[\vec{x}] = \exp \left[\frac{-(\vec{x} - \vec{v}_k^x)^T (\vec{x} - \vec{v}_k^x)}{2\sigma^2} \right] \quad (21)$$

where \vec{v}_k^x are the corresponding clusters for the inputs given by

$$\vec{v}_k^x = \sum_{x(\vec{p}) \in \text{cluster} < \text{CE:HSP SP} = "0.25"/> k} x(\vec{p}) \quad (22)$$

The weight vector for a hidden node equals

$$\vec{w}_k = \frac{\vec{v}_k^y}{\sum_{k=1}^K N_k \exp[-d(\vec{x}, \vec{v}_k^x)^2 / 2\sigma^2]} \quad (23)$$

where N_k is the number of input data in the cluster center k , the distance function is

$$d(\vec{x}, \vec{v}_k^x) = \sqrt{(\vec{x} - \vec{v}_k^x)^T (\vec{x} - \vec{v}_k^x)} \quad (24)$$

\vec{v}_k^y are the corresponding clusters for the outputs obtained by applying clustering technique of the input/output data that produces K cluster centers. The outputs of the hidden layer nodes are then multiplied with appropriate interconnection weights to produce the output of the GRNN. The configuration of the neural network model is determined by the nature of the problem to be solved. The number of input neurons is defined by the dimension of input vectors [27].

3.12. Ridge Regression (RR)

Ridge regression is a method from classical statistics that implements a regularized form of Least Squares Regression [34]. In its simplest form, ridge regression determines the parameter vector, $w \in \mathbb{R}^d$, of a linear model, $f(x) = \vec{w} \cdot \vec{x}$, by minimizing the objective function

$$W_{RR}(\vec{w}) = \frac{1}{2} \|\vec{w}\|^2 + \frac{\gamma}{N} \sum_{i=1}^N (y_i - \vec{w}x_i)^2 \quad (25)$$

The objective function used in ridge regression above implements a form of Tikhonov regularization of a sum-of-squares error metric, where γ is a regularization parameter controlling the bias-variance tradeoff. This corresponds to penalized maximum likelihood estimation of \vec{w} , assuming the targets have been corrupted by an independent and identically distributed (i.i.d.) sample from a Gaussian noise process, with zero mean and variance σ^2 , i.e.

$$y_i = \vec{w} \cdot \vec{x}_i + \varepsilon_i \quad \text{where } \varepsilon \sim \mathcal{N}(0, \sigma^2) \quad (26)$$

3.13. Total Least Squares Method (TLS)

Given an overdetermined set of m linear equations $Xw \approx d$ in n unknowns x , the LS method finds a solution x which

$$\text{minimizes } \|d - y\| \quad \text{subject to } xw = y \quad (27)$$

while the TLS method searches for a solution x which

$$\text{minimizes } \|[x; d] - [\hat{x}; y]\|_F \quad \text{subject to } xw = y \quad (28)$$

where F denotes the Frobenius norm [35].

Fig. 3 illustrates graphically the difference between the one-dimensional LS and TLS case. The dashed lines denote LS error and the solid lines perpendicular to line $b = xa$ denote TLS error. The LS minimizes the squared sum of the vertical distances, whereas the TLS method minimizes the squared sum of the perpendicular distances.

3.14. Partial Least Squares (PLS)

Partial Least Squares (PLS) is a wide class of methods for modeling relations between sets of observed variables by means of latent (not directly observed or measured) variables. It comprises of regression and classification task as well as dimension reduction techniques and modeling tools. The underlying assumption of PLS methods is that the observed data is generated by a system or process which is driven by a small number of latent variables [32]. PLS regression algorithm can build linear regression models and is useful in situations when the number of observed features is significantly greater than the number of observations and high multicollinearity among the features exists. Suppose the input features $X \subset \mathbb{R}^m$ and output targets $Y \subset \mathbb{R}$, PLS proposed by [31] uses a robust procedure, a Nonlinear Iterative Partial Least Squares (NIPALS) algorithm to solve a Singular Value Decomposition (SVD) problem.

3.15. Principal Component Regression (PCR)

The method of Principal Component Regression (PCR) consists of two estimation steps [36]. In the first step, A is estimated by means of principal components. That is, the p factors are obtained by minimizing the squared Frobenius norm of a matrix $\|X - \hat{X}\|^2$ under the restriction that \hat{X} has rank p . The squared Frobenius norm of a matrix is simply the sum of squares of all elements of the matrix. The X variables should be standardized to prevent scale effects. For instance, each column (variable) of X is scaled to have zero mean and unit norm. The estimates A can then be obtained from the singular value decomposition (SVD) of X . A more detailed explanation of the first step can be given as follows: Let $X = USV'$ be an SVD of X where the singular values in the matrix S are listed in decreasing order. Then $\hat{X} = U_p S_p V_p'$ where U_p and V_p consists respectively of the first p columns of U and V , and where S_p is the $p \times p$ diagonal matrix with the p largest singular values of X on the diagonal. If we define $W = V_p V_p'$ then it is easily checked that $\hat{X} = XW = XAB$ for any $k \times p$ matrix A and $p \times k$ matrix B such that $AB = W$. For instance, if we take $A = V_p S_p^{-1}$ and $B = S_p V_p'$, it follows that the vectors $F = XA$ satisfy $F'F = A'X'XA = I_p$ so that the p factors in F are scaled and mutually orthogonal. The factors F constructed in this way are the first p principal components at X . So, in PCR the parameter matrix A is estimated by minimizing $f_X(A, B) = \|X - XAB\|^2$. In the second step, the parameters α and β in $\hat{y}_{T+1} = \alpha + f_{T+1}\beta = \alpha + x_{T+1}A\beta$ are estimated by ordinary least squares (OLS), for given values of A . Let $F = XA$, the second step corresponds to minimizing $f_Y(\alpha, \beta) = \|y - \alpha - F\beta\|^2 = \|y - \alpha - XA\beta\|^2$.

3.16. Support Vector Regression (SMO)

The basic idea in Support Vector Regression (SMO) is to map the input data into a higher dimensional feature space via a non-linear mapping and then a linear regression problem is obtained and solved in this feature space [21]. In Support Vector Machines method, the regression function is approximated by the following function:

$$y = \sum_{i=1}^l w_i \phi_i(x) + b \quad (29)$$

where $\{\phi_i(x)\}_{i=1}^l$ are the features of inputs, $\{w_i\}_{i=1}^l$ and b are coefficients. The coefficients are estimated by minimizing the regularized risk function

$$R(C) = C \frac{1}{l} \sum_{i=1}^l L_\varepsilon(d_i, y_i) + \frac{1}{2} \|\vec{w}\|^2 \quad (30)$$

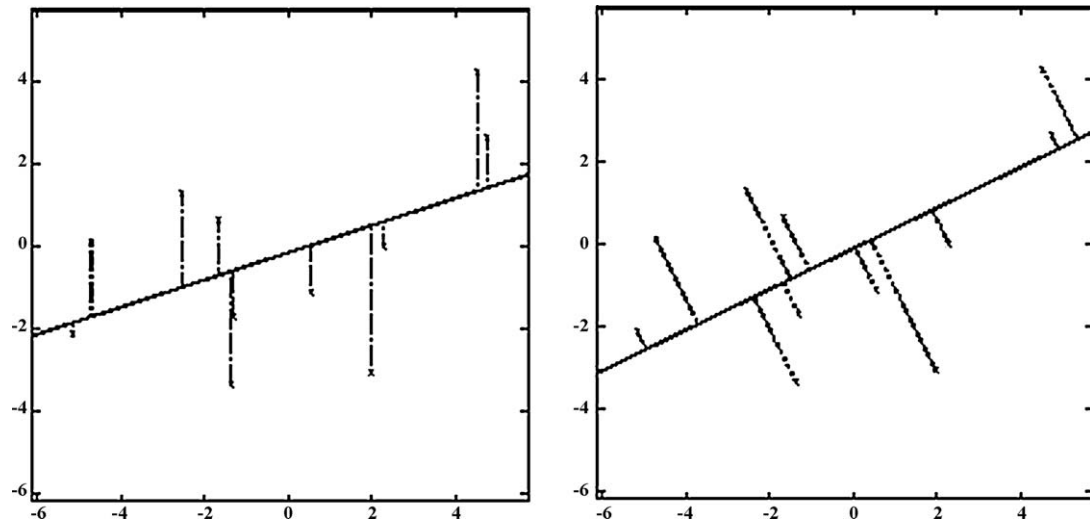


Fig. 3. LS versus TLS.

Table 3
Main Design Parameters Prediction Errors.

	<i>P</i>	<i>DWT</i>	<i>L_{BP}</i>	<i>B</i>	<i>T</i>	<i>V</i>	<i>Mean</i>
<i>LMS</i>	0.08(0.04)	0.14(0.07)	0.05(0.02)	0.05(0.02)	0.06(0.03)	0.15(0.07)	0.09
<i>LR</i>	0.06(0.01)	0.08(0.01)	0.05(0.01)	0.05(0.01)	0.06(0.02)	0.12(0.03)	0.07
<i>PR</i>	0.06(0.01)	0.08(0.01)	0.05(0.01)	0.05(0.01)	0.06(0.02)	0.12(0.03)	0.07
<i>RBF</i>	0.14(0.03)	0.14(0.04)	0.14(0.03)	0.14(0.02)	0.15(0.03)	0.17(0.04)	0.15
<i>SMO</i>	0.06(0.02)	0.09(0.03)	0.05(0.01)	0.05(0.01)	0.06(0.02)	0.12(0.02)	0.07
<i>SLR</i>	0.07(0.01)	0.08(0.02)	0.06(0.01)	0.06(0.02)	0.06(0.02)	0.13(0.03)	0.08
<i>M5P</i>	0.06(0.01)	0.03(0.01)	0.04(0.01)	0.04(0.01)	0.05(0.02)	0.11(0.02)	0.05
<i>M5R</i>	0.06(0.02)	0.04(0.14)	0.04(0.01)	0.04(0.02)	0.05(0.02)	0.11(0.02)	0.06
<i>Mean</i>	0.07	0.08	0.06	0.06	0.07	0.13	

where If $|d - y| \geq \varepsilon$ then $L_\varepsilon(d, y) = |d - y| - \varepsilon$, otherwise $L_\varepsilon(d, y) = 0$ and ε is a prescribed parameter.

3.17. M5 Model Trees (M5P)

M5 algorithm uses the following idea: split the parameter space into areas (subspaces) and build in each of them a local specialized linear regression model [20]. The splitting in Model Tree (MT) follow the idea used in building a decision tree, but instead of the class labels it has linear regression functions at leaves, which can predict continuous numeric attributes. Model trees generalize the concepts of regression trees, which have constant values at their leaves. So, they are analogous to piecewise linear functions (and hence non-linear). Model trees learn efficiently and can tackle tasks with very high dimensionality – up to hundreds of attributes. The major advantage of model trees over regression trees is that model trees are much smaller than regression trees, the decision strength is clear, and the regression functions do not normally involve many variables. Fig. 4 illustrates the M5 model trees mechanism.

3.18. M5 Model Rules (M5R)

Generates a decision list for regression problems using separate-and-conquer. In each iteration it builds a model tree using M5 and makes the “best” leaf into a rule [19].

4. Experimental results

This study has two main objectives. First, predictions are made to predict each of six main parameters used in chem-

ical tanker pre-design given the other parameters. Second, a more detailed investigation is conducted on the estimation of the most important parameter, namely the main engine power (*P*).

The used algorithms can be divided into two categories as linear model generators and non-linear model generators. The best performed algorithms over our datasets (M5P, M5R, GDA) are non-linear model generators for our datasets. This results say the relation between our attributes are non-linear. The root MSE is used as only performance measure between algorithms since the training time, memory requirements are meaningless in such a small datasets as ours.

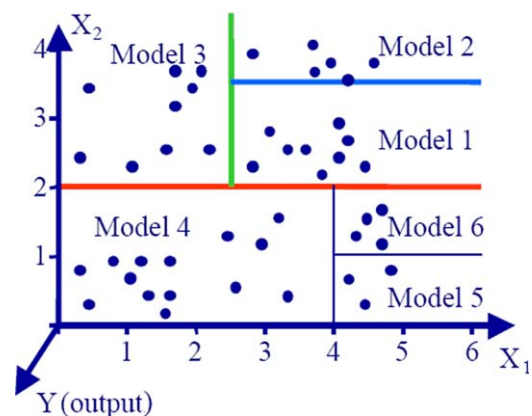


Fig. 4. Splitting the input space by M5 model tree algorithm; each model is a linear regression model.

Table 4

The statistical significance of Algorithms' Prediction Errors, formed as the number of (win/tie/lost) out of 6 datasets.

	LMS	LR	PR	RBF	SMO	SLR	M5P
LR	(1/5/0)						
PR	(1/5/0)	(0/6/0)					
RBF	(0/2/4)	(0/0/6)	(0/0/6)				
SMO	(1/5/0)	(0/6/0)	(0/6/0)	(6/0/0)			
SLR	(0/5/1)	(0/3/3)	(0/3/3)	(6/0/0)	(0/3/3)		
M5P	(3/3/0)	(3/3/0)	(4/2/0)	(6/0/0)	(4/2/0)	(6/0/0)	
M5R	(3/3/0)	(2/4/0)	(2/4/0)	(6/0/0)	(4/2/0)	(4/2/0)	(0/6/0)

4.1. Predictions of Main Design Parameters

In Table 3, 6 main design parameters prediction experiments are summarized. For the generalization of results, 10 times 10-fold cross validation is applied. For each fold, training and test sets are composed of 99 and 15 samples, respectively. The values are the means of 100 (10 × 10) experiment's root MSE (RMSE) and standard deviations in parenthesis.

$$\begin{aligned}
 LM1 : DWT &= 0.2041 \times L_{BP} + 0.1687 \times B + 0.0649 \times T - 0.0386 \times V + 0.0233 \times P - 0.02272 \\
 LM2 : DWT &= 0.3016 \times L_{BP} + 0.3335 \times B + 0.1998 \times T - 0.0386 \times V - 0.0597 \times P - 0.1224 \\
 LM3 : DWT &= 0.435 \times L_{BP} + 0.2884 \times B + 0.5183 \times T - 0.1838 \times V + 0.0628 \times P - 0.2958 \\
 LM4 : DWT &= 0.7634 \times L_{BP} + 0.2884 \times B + 0.6132 \times T - 0.1632 \times V + 0.0628 \times P - 0.643 \\
 LM5 : DWT &= 0.5634 \times L_{BP} + 0.2884 \times B + 0.6318 \times T - 0.1632 \times V + 0.0628 \times P - 0.4511
 \end{aligned}$$

For the explanations of the method and the abbreviations see Table 2. The parameters estimated at best on average are L_{BP} and B with a RMSE of 0.06. The worst guess on average is of V with a RMSE of 0.13.

To be able to assess the statistical significance of the differences, Paired T -tester is applied with 0.05 significance value. In Table 4, each algorithm is tested with all other algorithms over 6 datasets. The values in table cell are formed as the number of wins, the number of tie and the number of lost, respectively.

According to Tables 4 and 3, the most successful method is M5P with RMSE of 0.06 and the most win number (26 out of 42 comparisons). Also, there is no significant difference between M5P and M5R performances (see the intersection cell of M5P and M5R algorithms

$$\begin{aligned}
 LM1 : L_{BP} &= 0.0924 \times B - 0.158 \times T + 0.0684 \times V + 0.1996 \times PKW + 0.9669 \times DWT + 0.1948 \\
 LM2 : L_{BP} &= 0.5375 \times B + 0.0502 \times T + 0.0574 \times V + 0.1117 \times PKW + 0.8781 \times DWT + 0.0232 \\
 LM3 : L_{BP} &= 0.232 \times B + 0.2262 \times T + 0.3785 \\
 LM4 : L_{BP} &= -0.1517 \times B - 0.1047 \times V + 0.1778 \times DWT + 0.9828 \\
 LM5 : L_{BP} &= 0.39 \times DWT + 0.6319
 \end{aligned}$$

at Table 4). M5P and M5R algorithms have another advantage over other algorithms, since they generate easy to understand rules. The least successful method is RBF with a RMSE of 0.15 and 0 win number. For each individual parameter, the method approximates best and the associated rule are given as follows.

4.1.1. Prediction of Main Engine Power (P)

The best method is M5P with a RMSE of 0.06. The rule associated is as follows.

$$\begin{aligned}
 L_{BP} &\leq 0.561 : \\
 L_{BP} &\leq 0.386 : LM1 \\
 L_{BP} &> 0.386 : LM2 \\
 L_{BP} &> 0.561 : \\
 L_{BP} &\leq 0.672 : LM3 \\
 L_{BP} &> 0.672 : LM4
 \end{aligned}$$

$$\begin{aligned}
 LM1 : P &= 0.1311 \times DWT + 0.1431 \times L_{BP} + 0.3864 \times B + 0.2289 \times V - 0.0197 \\
 LM2 : P &= 0.6576 \times DWT + 0.1552 \times L_{BP} + 0.0529 \times B + 0.3799 \times V - 0.0211 \\
 LM3 : P &= 0.1167 \times DWT + 0.4946 \times L_{BP} + 0.0312 \times B + 0.1894 \times V + 0.1154 \\
 LM4 : P &= 0.0979 \times DWT + 0.4256 \times L_{BP} - 0.0837 \times B + 0.1615 \times V + 0.3252
 \end{aligned}$$

4.1.2. Prediction of Dead Weight Tone (DWT)

The best method is M5P with a RMSE of 0.0293. The rule associated is as follows.

$$\begin{aligned}
 T &\leq 0.577 : \\
 L_{BP} &\leq 0.344 : LM1 \\
 L_{BP} &> 0.344 : LM2 \\
 T &> 0.577 : \\
 L_{BP} &\leq 0.853 : LM3 \\
 L_{BP} &> 0.853 : \\
 T &\leq 0.919 : LM4 \\
 T &> 0.919 : LM5
 \end{aligned}$$

4.1.3. Prediction of Length Between Perpendiculars (L_{BP})

The best method is M5R with a RMSE of 0.0369. The rule associated, which is given below, should be applied in this order.

$$\begin{aligned}
 PKW &\leq 0.537 : \\
 DWT &> 0.092 : LM1 \\
 B &\leq 0.369 : LM2 \\
 B &\leq 0.633 : LM3 \\
 DWT &> 0.709 : \\
 B &> 0.881 : LM4 \\
 LM5
 \end{aligned}$$

4.1.4. Prediction of Breadth (B)

The best method is M5P with a RMSE of 0.0378. The rule associated is

$$\begin{aligned}
 DWT &\leq 0.254 : \\
 DWT &\leq 0.05 : \\
 T &\leq 0.086 : LM1 \\
 T &> 0.086 : \\
 DWT &\leq 0.046 : LM2 \\
 DWT &> 0.046 : LM3 \\
 DWT &> 0.05 : LM4 \\
 DWT &> 0.254 : LM5
 \end{aligned}$$

$$\begin{aligned}
 LM1 : B &= 0.4468 \times T - 0.0281 \times V + 0.118 \times P + 0.5456 \times DWT + 0.2314 \times L_{BP} + 0.0129 \\
 LM2 : B &= 0.2353 \times T - 0.0333 \times V + 0.1395 \times P + 0.8777 \times DWT + 0.2314 \times L_{BP} + 0.0223 \\
 LM3 : B &= 0.2353 \times T - 0.0333 \times V + 0.1395 \times P + 0.9265 \times DWT + 0.2314 \times L_{BP} + 0.0222 \\
 LM4 : B &= 0.0736 \times T + 0.096 \times P + 0.9138 \times DWT + 0.4332 \times L_{BP} + 0.0951 \\
 LM5 : B &= -0.1581 \times T - 0.0864 \times P + 0.6948 \times DWT + 0.1522 \times L_{BP} + 0.3851
 \end{aligned}$$

Table 5

The main engine power (P) prediction errors.

P prediction success order with GDM trials	B	T	L_{BP}	DWT	V
P prediction RMSEs	0.1208	0.1263	0.2281	0.3147	0.4826
Parameter order according to correlation with P	L_{BP}	B	T	DWT	V

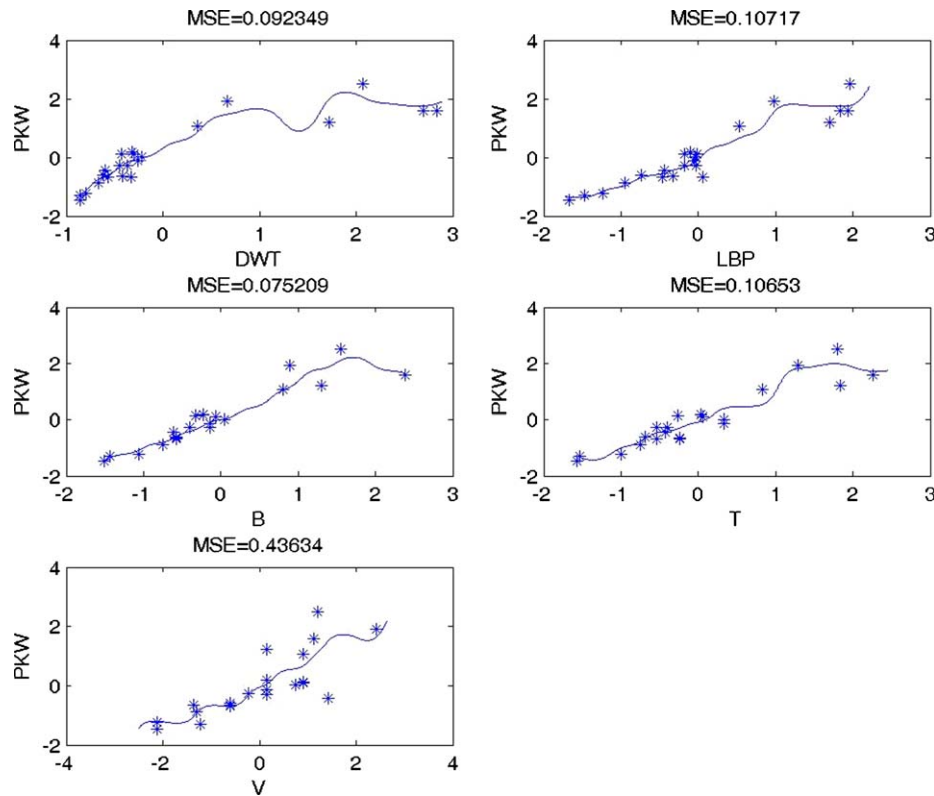


Fig. 5. Main power predictions using only 1 parameter.

4.1.5. Prediction of Draught (T)

The best method is M5P with a RMSE of 0.0477. The rule associated is

$DWT \leq 0.169$:
 $P \leq 0.145$:
 $DWT \leq 0.011$: LM1
 $DWT > 0.011$: LM2
 $P > 0.145$: LM3
 $DWT > 0.169$:
 $DWT \leq 0.365$: LM4
 $DWT > 0.365$:
 $DWT \leq 0.926$:
 $DWT \leq 0.56$: LM5
 $DWT > 0.56$:
 $B \leq 0.793$: LM6
 $B > 0.793$: LM7
 $DWT > 0.926$: LM8

LM1 : $T = 0.0554 \times V + 0.0169 \times P + 1.2989 \times DWT + 0.0367 \times L_{BP} + 0.1966 \times B + 0.0163$
 LM2 : $T = 0.0543 \times V + 0.0169 \times P + 1.4804 \times DWT - 0.0003 \times L_{BP} + 0.1966 \times B + 0.0398$
 LM3 : $T = 0.0847 \times V + 0.0169 \times P + 1.4787 \times DWT - 0.0909 \times L_{BP} + 0.1345 \times B + 0.0898$
 LM4 : $T = 0.0631 \times V + 0.1694 \times P + 0.9042 \times DWT - 0.4549 \times L_{BP} + 0.0487 \times B + 0.3714$
 LM5 : $T = 0.0313 \times V + 0.1210 \times P + 1.3274 \times DWT - 0.0595 \times L_{BP} + 0.1743 \times B + 0.3491$
 LM6 : $T = 0.0323 \times V + 0.1242 \times P + 0.3274 \times DWT - 0.0595 \times L_{BP} + 0.1389 \times B + 0.384$
 LM7 : $T = 0.0323 \times V + 0.1490 \times P + 0.3274 \times DWT - 0.0595 \times L_{BP} + 0.1407 \times B + 0.3604$
 LM8 : $T = 0.0643 \times V + 0.0602 \times P + 0.4246 \times DWT - 0.0595 \times L_{BP} + 0.2149 \times B + 0.2834$

4.1.6. Prediction of Ship Speed (V)

The best method is M5P with a RMSE of 0.1123. The rule associated is

$P \leq 0.234$: LM1
 $P > 0.234$: LM2

LM1 : $V = 2.0053 \times P - 2.6391 \times DWT - 0.5638 \times B + 0.973 \times T + 0.0825$
 LM2 : $V = 0.7921 \times P - 0.1118 \times DWT - 0.3631 \times B + 0.517 \times T + 0.3773$

4.2. Detailed Analysis of Main Engine Power (P) Predictions

Due to the fact that the estimation of the main power is of utmost importance, the parameter is examined in detail and the estimation is effectuated on several different toolboxes. The first step was estimation trials on neural network toolbox to find out which individual parameter has the greatest influence on main engine power. Fig. 5 shows the relation of P with each of other parameters using the GDM method.

Table 6

The best main engine power (P) prediction performances using 5 design parameters.

Method	RMSE
LM	2.4605
GDA	0.0454
GDM	0.0521
GRNN	0.0775
MLR	0.0925
OLS	0.0925
RR	0.0884
TLS	0.6133
PCR	0.0932
PLS	0.0928

Table 5 combines the success order of the methods using 1-parameter trials and the success order according to correlation results explained in Table 1. Both of the two orders can be seen fairly in parallel.

Trials with 1-parameter are followed by trials including 2, 3, 4 and 5 parameters together. Trials with 5-parameters, namely all remaining parameter experiments has given the best approximation. The results for the trials with 5-parameters are shown in Table 6.

The values given in Table 6 are the best results of each method with different parameter configurations. The most successful method is GDA with RMSE of 0.0454, which is better than M5P with a RMSE of 0.06 on WEKA toolbox.

5. Conclusion

In naval engineering, the conventional methods use several parameters (although some of which require tedious work and costly experiments) to calculate the engine power, the water displacement and other pre-design parameters [30]. As an alternative methodology, this study aims to estimate main design parameters by the help of several machine learning algorithms. The two main objectives of this study are discovery of the best/worst predictable parameter among all design parameter estimation. The second one is to find out which parameter is the most influent on and which machine learning method is the most successful in the estimation of the most important chemical tanker pre-design parameter, main engine power (P). In order to find out the best approximation method, 6 regression methods included in WEKA toolbox are used. To estimate each individual parameter, experiments are made using all of the five remaining parameters and the rules are given in details and trials for best approximations evaluated. The experiments show that the best approximated parameter is the length (L_{BP}) while the worst one is the speed (V). The most successful method is Model Trees (M5P). In order to find out the main power using 1-parameter only, several regression methods in MATLAB neural network toolbox (NNET) are used. L_{BP} emerges as the parameter that defines best the PKW. Experiments using 2, 3, 4 and 5 parameters at a time to estimate PKW are realized using NNET. The experiments show that the best approximation comes out when all the 5 remaining parameters are used collectively. Finally, 5 parameter approximations are evaluated both with NNET regression toolbox and PLS toolbox. The best method is found to be GDA for all toolboxes.

In this study, it is shown that all the main pre-design parameters except the speed can be predicted accurately. And it is proposed that the expensive and time consuming conventional ship model experiments can be replaced by the computational predictive methods.

Finally, we would like to thank Assist. Prof. Aykut SAFA for their thoughtful comments.

References

- [1] A.D. Alkan, K. Gulez, H. Yilmaz, Design of a robust neural network structure for determining initial stability particulars of fishing vessels, *Ocean Engineering* 31 (2004) 761–777.
- [2] M.F. Amasyali, M. Bal, U.B. Celebi, S. Ekinici, U.K. Boyaci, Comparison of different neural networks performances on motorboat datasets, *LNCS Volume 4263* Springer, in: 21st Intl. Symposium on Computer and Information Sciences ISCIS-2006, Istanbul, Turkey, 2006.
- [3] M. Bal, M.F. Amasyali, U.B. Celebi, S. Ekinici, Using machine learning methods in the computation of engine power of motorboats and the comparison of these methods, in: *Int. Conference on Modelling and Simulation-AMSE'06*, Konya, Turkey, 2006.
- [4] M.R. Haddara, M. Wishahy, An investigation of roll characteristics of two full scale ships at sea, *Ocean Engineering* 29 (2002) 651–666.
- [5] H. Kuo, L. Wu, J. Chen, Neural-fuzzy fault diagnosis in a marine propulsion shaft system, *Journal of Materials Processing Technology* 122 (2002) 12–22.
- [6] A.B. Mahfouz, Identification of the nonlinear ship rolling motion equation using the measured response at sea, *Ocean Engineering* 31 (2004) 2139–2156.
- [7] G.N. Roberts, R. Sutton, A. Zirilli, A. Tiano, Intelligent ship autopilots—a historical perspective, *Mechatronics* 13 (2003) 1091–1103.
- [8] A. Zubaydi, M.R. Haddara, A.S.J. Swamida, Damage identification in a ship's structure using neural networks, *Ocean Engineering* 29 (2002) 1187–1200.
- [9] M.M. Islam, R.H. Khondoker, C.M. Rahman, Application of artificial intelligence techniques in automatic hull form generation, *Ocean Engineering* 28 (2001) 1531–1544.
- [10] N.I. Xiros, N.P. Kyrtatos, A neural predictor of propeller load demand for improved control of diesel ship propulsion, in: *Proceedings of the 15th IEEE International Symposium on Intelligent Control (ISIC)*, 2000, pp. 321–326.
- [11] C. Chen, J. Lee, M. Lin, Classification of underwater signals using wavelet transforms and neural networks, *Mathl. Comput. Model.* 27 (2) (1998) 47–60.
- [12] W.S. Filho, J.M. Seixas, L.P. Calôba, Averaging spectra to improve the classification of the noise radiated by ships using neural networks, in: *Neural Networks, Proceedings. Sixth Brazilian Symposium*, 2000, pp. 156–161.
- [13] J.R. Fonseca, L. Correia, Identification of underwater acoustic noise, pp. 11/597–602, in: *OCEANS'94, Oceans Engineering for Today's Technology and Tomorrow's Preservation*, Proceedings, vol. 2, 1994.
- [14] M. Farrokhrooz, M. Karimi, Ship noise classification using probabilistic neural network and AR model coefficients, in: *IEEE, Oceans 2005-Europe*, vol. 2, 2005, pp. 1107–1110.
- [15] J. Chen, H. Li, S. Tang, J. Sun, A SOM-based probabilistic neural network for classification of ship noises, in: *Proceedings of IEEE International Conference Communications, Circuits and System and WestSino Expositions*, vol. 2, 2002, pp. 1209–1212.
- [16] G. Yang, X. Wu, Q. Zhang, Y. Chen, Fault diagnosis of ship main power system based on multi-layer fuzzy neural network, in: *IEEE, Proceedings of the 6th World Congress on Intelligent Control and Automation*, June-21–23, Dalian, China, 2006, pp. 5713–5717.
- [17] J. Xiao, X. Wang, M. Bao, The modeling and simulation of ship main engine based on neural network, in: *Intelligent Control and Automation*, Proceedings of the 4th World Congress, vol. 1, 2002, pp. 123–125.
- [18] Y. Zhan, B. Wan, X. Wang, Y. Hu, A simulation model for the main engine of the modern container ship, pp. 2996–3002, in: *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, Shanghai, 2004.
- [19] M. Hall, G. Holmes, E. Frank, Generating rule sets from model trees, in: *Proceedings of the Twelfth Australian Joint Conference on Artificial Intelligence*, Springer-Verlag, Sydney, Australia, 1999, pp. 1–12.
- [20] B. Bhattacharya, D.P. Solomanite, Neural network and M5 model trees in modelling water level-discharge relationship, *Neurocomputing* 53 (2005) 381–396.
- [21] W. Wang, Z. Xu, A heuristic training for support vector regression, *Neurocomputing* 61 (2004) 259–275.
- [22] Akaike, Hirotugu, A new look at the statistical model identification, *IEEE Transactions on Automatic Control* 19 (6) (1974) 716–723.
- [23] D.E. Rumelhart, J.L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press, 1986.
- [24] R.P. Lippmann, An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, 1987.
- [25] G. Corsini, M. Diani, R. Grasso, M. De Martino, P. Mantero, S.B. Serpico, Radial basis function and multilayer perceptron neural networks for sea water optically active parameter estimation in case waters: a comparison, *International Journal Remote Sensing* 24 (20) (2003).
- [26] Y. Chen, B.M. Wilamowski, TREAT: a trust-region-based error-aggregated training algorithm for neural network, in: *Proceedings of the IEEE International Joint Conference on Neural Networks-IJCNN 2002*, Honolulu, Hawaii, 2002.
- [27] I. Popescu, A. Kanatas, P. Constantinou, I. Naforita, Applications of general regression neural networks for path loss prediction, in: *Proceedings of the International Workshop Trends and Recent Achievements in Information Technology*, Cluj Napoca, 2002.
- [28] D.F. Specht, A generalized regression neural network, *IEEE Transactions on Neural Networks* 2 (November) (1991) 568–576.
- [29] Z.C. Lin, W.J. Wu, Multiple linear regression analysis of the overlay accuracy model, *IEEE transactions on Semiconductor Manufacturing* 12 (May (2)) (1999) 229–237.
- [30] R. Baykal, A.C. Dikili, *Ship Resistance and Engine Power*, I.T.U, ISBN 975-561-196-7, Istanbul, 2002 (in Turkish).

- [31] L. Eriksson, E. Johansson, N. Kettaneh-Wold, S. Wold, Multi- and megavariate data analysis: principles and applications, Umetrics, Umea, 2001.
- [32] R. Rosipal, N. Kramer, Overview and Recent in Partial Least Squares, *Lecture Notes in Computer Science*, vol. 3940, Springer Verlag, Berlin, 2006, pp. 34–51.
- [33] E.S. Cheng, H. Yang, S. Bs, Orthogonal least squares learning algorithm with local adaptation process for the radial basis function networks, *IEEE Signal Processing Letters* 3 (1996) 253–255.
- [34] G.C. Cawley, N.L.C. Talbot, Reduced rank kernel ridge regression, *Neural Processing Letters* 16 (2002) 293–302.
- [35] M. Xia, E. Saber, G. Sharma, M. Tekalp, End-to-end color printer calibration by total least squares regression, *IEEE Transactions on Image Processing* 8 (1999) 700–716.
- [36] C. Heij, P.J.F. Groenen, D. Van Dijk, Forecast comparison of principal component regression and principal covariate regression, *Computational Statistics and Data Analysis* 51 (2007) 3612–3625.