Abstract

The maturity of Reynolds-averaged Navier Stokes computational fluid dynamics (RANS-CFD) packages offers the ready assessment of the hydro-mechanic performance of a wind-assisted commercial ship. However, these simulations require intensive computational resources and complex software to generate results. In the hull optimisation context, individual adjustments to parameters require discrete processing that may take hours or days to process. To expedite results under different hull designs, or in the context of bulk analysis of multiple vessels in a fleet, machine learning models are trained on a database of RANS-CFD for a systematic hull series. The model successfully trained on the RANS-CFD results with over 94% accuracy for 60 different hull variations. Using the trained model to evaluate new runs on individual hulls showed over 86% accuracies, allowing for rapid, accurate reproduction of vessel response for generic wind-assist hulls which can be quickly evaluated under a wide variety of sailing conditions.

Machine Learning Based Hydro-mechanic Modeling for Sailing Commercial Ships

Nico van der Kolk[*]
Blue Wasp Marine
nvanderkolkbluewaspmarine.com

Brian S. Freeman
Lakes Environmental
brian.freeman@weblakes.com

I. Introduction

Wind assisted ship propulsion stands apart among available technologies for the energy transition in commercial shipping. A wind-hybrid vessel promises to deliver substantial fuel savings, a result that has been reported by several researchers in recent years [? ? ? ?]. This promise of substantial reductions in emissions, for both local pollutants and for greenhouse gases, is achievable in the near-term.

Technological readiness of wind assist concepts is not the barrier to broader market uptake. Several viable concepts for wind propulsors are commercially available. Rather, it is a lack of experience with industrialized sailing and unwillingness to take risk as an 'early adopter'. The further development of this promising technology is hampered by a poor understanding of the interaction effects between wind propulsors and the hydromechanics of commercial ships. For the ship owner or operator, this lack of experience with industrial sailing introduces uncertainty in a profoundly risk-averse sector. For the regulator who wishes to promote the uptake of sustainable technologies, the knowledge gap raises the spectre of misdirected policies that fail to advance wind assisted ship propulsion as a viable component of the energy transition. In fact, wind propulsion is one of the only interventions in the maritime shipping sector that promises significant reductions in greenhouse gas emissions in the near term. Furthermore, besides the simple arithmetic of fuel savings and limiting exposure to increasingly volatile fuel prices, wind-assistance raises the possibility of engaging with an activist consumer class and potentially increasing the perceived value of shipped products.

i. Wind-assist vessel model

The reliable performance prediction of a wind-assisted ship is a necessary prerequisite for any sound economic and environmental evaluation of these concepts. Recognizing a general lack of understanding about the physics of wind-assist vessels, and the need for an assessment tool to facilitate the further development of this promising technology, a performance prediction tool is being developed by the Sail Assist research group. A quick yet reliable way to assess the performance of different wind-assisted ships will be useful throughout the design process. In early stages, when the user wants to explore several different designs, the in-built force models under development at Delft University of Technology may be used. In a more advanced design phase, the user can input data obtained by dedicated simulations or experiments. In this case the modeling tool is used purely as a solver to obtain very detailed results for a specific design.

The fundamental task of the performance prediction tool is to balance the aerodynamic and hydrodynamic forces acting on a wind-assisted ship to arrive at a sailing equilibrium. This is done within an optimization routine that minimizes the engine delivered thrust, $T(1-t)$, while maintaining a prescribed vessel speed. Alternative optimization routines

[*] Corresponding author

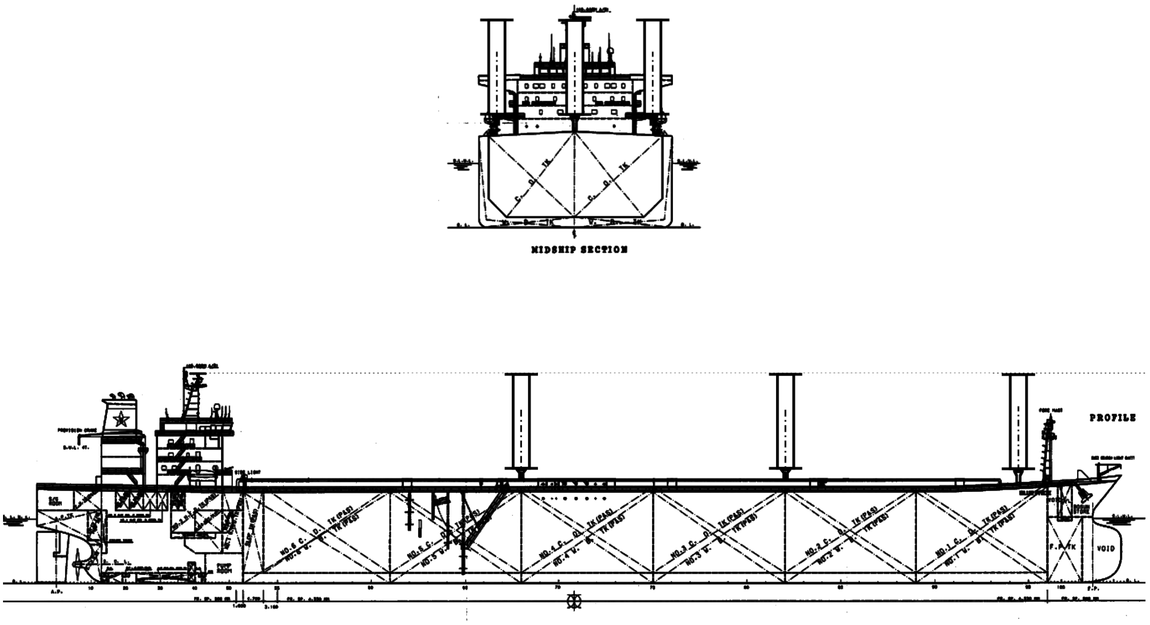


Figure 1: *80,000 DWT Panamax bulker with three 30-meter Flettner rotors.*

may include pure sailing, delivering nominal engine power and maximizing speed, or maintaining a minimum speed. The program will calculate the performance of the wind-assisted ship for a specified range of true wind angle and true wind speed.

P_{WASP} considers the wind-assist propulsive power against any power needed to operate the wind propulsor. It is defined as:

$$P_{\mathrm{WASP}} = \frac{V_{\mathrm{ref}}}{\eta_{\mathrm{T}}}\left(X_{\mathrm{Aero}} - \Delta R_{\mathrm{Sailing}}\right) - \frac{P_{\mathrm{Rotor}}}{\eta_{\mathrm{Gen}}} \tag{1}$$

A design can be evaluated and improved based on these polar diagrams, or this information can be passed to a weather routing program or used to define the control systems of the wind propulsor. Finally, vessel performance with wind assist operation can be optimized, and studied for environmental impact and economic investment evaluations.

II. Methodology

i. Modeling for hydro-mechanic response

The ship adopts a heel and leeway angle to support the sail-plan. This combination places the hull—which is otherwise optimized for quite specific and symmetric operating conditions—oblique to the mean flow in the *sailing condition*. The normal wave field produced as part of the wake of this ship will be superimposed on the pressure distribution arising from the sideforce production by the hull, along with the Munk yawing moment [**?**]. Finally, a vessel heeling angle will bring the vessel "shoulders" closer to the free surface, leading to a further distortion of the wave system. The pressure resistance for a sailing ship will therefore vary with heel and leeway angles, alongside the ordinary Froude number dependency.

The sailing performance for wind-assisted ships is synonymous with maneuvering forces for the steady drift condition, including vessel heel angle. A right hand rule is adopted

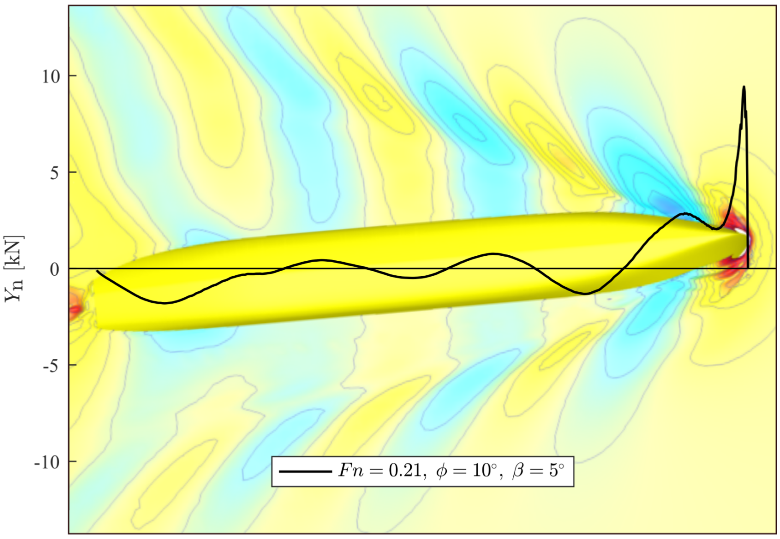


Figure 2: *Distribution of hydrodynamic sideforce, showing wave system (viewpoint is below the ship). Hull # 1, leeway angle $\beta = 5°$, heel angle $\phi = 10°$. As the vessel heels, the fore and aft shoulders are brought close to the free surface, causing flow constriction and acceleration. This effect is especially pronounced at large heel angles (ϕ=10° is normally adopted as operational limit for manned vessels).*

with the z axis pointed down. All rotations are performed about midship at the calm water line. Forces and moments are presented in flow-aligned coordinates, $< x, y >$, with the suitable transformation as in Figure 3.

The distribution of hydrodynamic sideforce (Sway) along the hull was extracted from simulation results. The field results for fluid pressure **P** and shear stress **S** are projected in the flow-normal direction (n_Y) and integrated on segments of the hull (as in Figure 3).

$$Y_n = \int_{\xi}^{\xi+d\xi} \int_{z=0}^{T_n} (\mathbf{P} + \mathbf{S}) \cos \beta dz d\xi \qquad (2)$$

i.1 Sailing efficiency

For modeling of wind-assist vessels, the resistance increase due to the sailing condition (heel angle ϕ, and leeway angle β) is expressed as an effective draft T_e. This quantity is related to the slope of a linear fit through data for several heel and leeway angles (bottom-left in **??**).

Following theories for low-aspect planforms [**? ?**], this induced drag may be significant for commercial ships, meaning that the thrust delivered by a wind propulsor might well be overwhelmed by this increase in re-

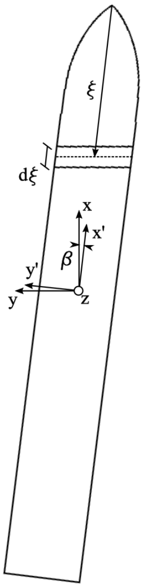


Figure 3: *Coordinate system for vessel hydro-mechanic response (viewpoint is below the ship).*

sistance. Though the flow mechanisms only vaguely resemble the Prandtl lifting-line and the associated derivation for the induced drag [**?**], the accounting for energy loss in shed vorticity is especially relevant for the present application. Following the analysis of sailing yachts by **?**], the resistance increase due to sideforce production is modeled as an effective draft, T_e [**?**], which is a metric for the sailing efficiency of the hull. The expression is derived from the lifting-line theory of wings. In non-dimensional form:

$$C_X = \frac{1}{\pi AR_{\text{eff}}} C_{\text{Y}}^2 + C_{XO} \tag{3a}$$

$$T_{\text{e}} = \sqrt{\frac{TLAR_{\text{eff}}}{2}} \tag{3b}$$

Forces and moments are non-dimensionalized using the dynamic pressure $q = \frac{1}{2}\rho U^2$ following the maneuvering convention. A linear and thrid order term are assumed for β, and a linear term for ϕ. The AR_{eff}, as in Equation (3a) is derived from a (linear) curve fit. A non-dimensional form for the effective draft T_{e} is made with the vessel draft: T_{e}/T Equation (3b), providing a convenient metric for the sailing efficiency of the hull. Some difficulty arises on account of the non-linear response of the commercial ship hull.

i.2 Vessel course keeping ability

The second quantity of principal interest for this modeling for wind-assist vessels is the center of effort for the distribution of lateral force, also known as the center of lateral resistance. The position of the CLR is determined as the quotient of the yaw moment and the sideforce.

$$CLR = C_{\text{N}}/C_{\text{Y}} \tag{4}$$

Again, body forces are non-dimensionalized with the dynamic pressure q.

$$C_{\text{X}} = X_{\beta\beta}\beta^2 + X_{\phi}\phi \tag{5}$$

$$C_{\text{Y}} = Y_{\beta}\beta + Y_{\beta\beta\beta}\beta^3 + Y_{\phi}\phi \tag{6}$$

$$C_{\text{N}} = N_{\beta}\beta + N_{\phi}\phi \tag{7}$$

Behavior for the CLR as function of leeway angle β is shown in Figure 4 for several appendage configurations: the bare hull (un-appended), the nominal appended hull design with rudder set for zero degrees, a short bilge keel in the most forward position, and a long bilge keel occupying the full length of the parallel midbody. The CLR is given for leeway angles β of three, six, and nine degrees.
First, observe the CLR for the un-appended hull, which lies more than half a ships-length ahead of the bow (shown only for nine degrees leeway), a consequence of the stronger development of the yawing momentlinear with leeway anglecompared against the sideforce, which includes a significant higher-order dependency for leeway angle. The CLR moves aft as the leeway angle increases, an effect that is driven by an increase in flow separation along the bilges. This effect is manifest as a rising contribution for the higher-order sideforce term in the sway equation, and an attenuation of the Munk moment for the yaw equation by flow separation along the vessel aft-body. Yaw balance, achieved by aligning the aerodynamic center of the wind propulsors with the hydrodynamic center (CLR) of the hull [**?**], is impossible.

The estimation of the hydrodynamic derivatives follows from analysis of results from the database of full-scale simulation results, as described in [**? ?**], for the DWA. This approach to vessel modeling follows the same methodology as **? ? ? ? ?**], in the maneuvering and sailing fields.

i.3 Reynolds-averaged Navier Stokes simulation

Reynolds-averaged Navier Stokes computational fluid dynamics (RANS-CFD) and other numerical modeling packages are often used during the development of performance predictions for commercial ships to estimate sailing performance [**? ? ?**]. These methods require intensive computational resources and complex software to generate results, thereby

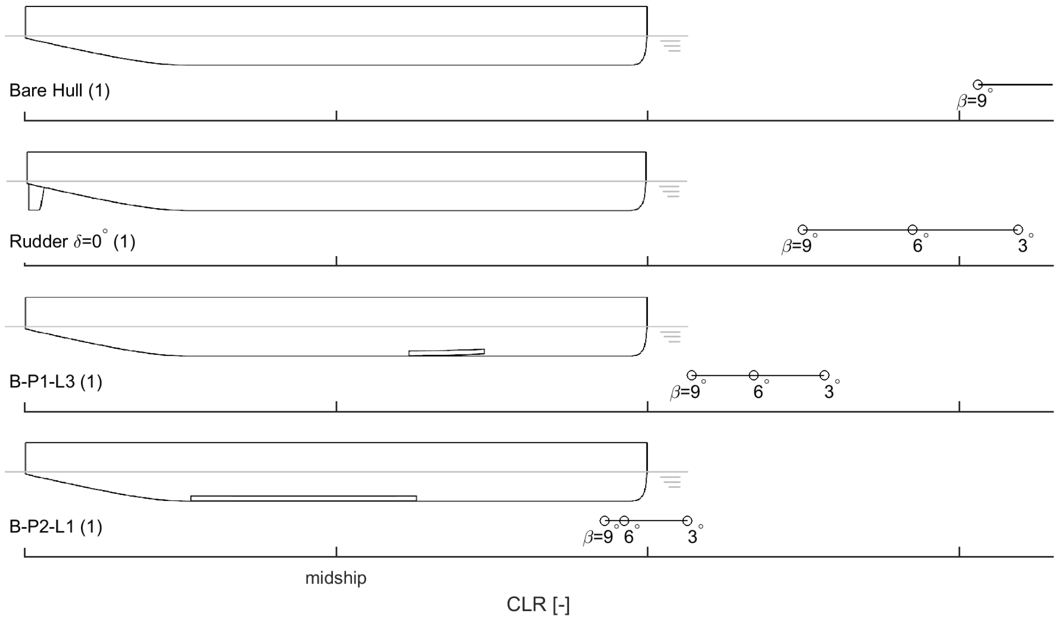


Figure 4: *Vessel CLR for several appended hulls. Experimental result.*

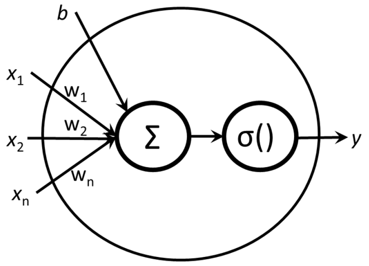
limiting the number of scenarios, hull variations, and operating conditions a specific design can be evaluated under. Also, the computational effort may exceed what is appropriate under varying modeling contexts; i.e. global energy spectra analysis using database of operating vessels as under scenario analyses, or specific hull optimization routines. Reliable performance predictions are essential in new ship design and when considering modifications to existing hulls in order to improve operability and sailing efficiency.

A full scale simulation method is adopted to resolve essential Reynolds scaling difficulties, as described in [**?**]. The full-scale simulation method for the production runs is designed for the assessment of hull geometry variants of the Delft Wind-Assist Series, an extensive series of wind-assist ship hulls. Considering the volume of work, to be done at full scale, a premium must be placed on economical simulations–precluding near-wall modeling or elaborate turbulence models. The RANS equations are solved with the ISIS-CFD flow solver, developed at École Centrale Nantes and commercialized by Numeca International. The ISIS-CFD flow solver is an in-compressible unsteady RANS method. The unstructured spatial discretization for the transport equations is based on the finite volume method. Free-surface flows are simulated with a conservation equation for the mass fraction. A detailed description of the solver is provided in [**? ? ? ?**].

It is understood that flow around the hull will include large anisotropic vortices that will play a key role in the sailing performance of the hull. Turbulence is modeled with the Explicit Algebraic Stress Model (EASM), providing a balance between the Boussinesq-modeling and the modeling of Reynolds stresses and giving a more physical approach while remaining viable within the scope of work and for the computational resources available. The evaluation of convective terms in the momentum equation and the turbulent stresses is performed with a blended upwind/central scheme based on the local Courant number (Co). The solution for the free surface is determined following the volume of fluid method using an interface compression algorithm that is likewise dependent on the local Courant number. Complete documentation of the ISIS-CFD (FINE/Marine) solver is available [**?**]. Production simulation work was carried out at the Netherlands high performance computing cluster [**?**].

ii. Numerical modeling with machine learning

Machine learning is a generic term that covers a broad range of analytical processes including linear regression. Machine learning uses datasets to train software to recognize patterns or predict outcomes by updating parameters within an algorithm that minimizes the error associated with the data. The error could be the difference between the expected output versus the calculated output or simply be the shortest distance between a set of point. The most common type of machine learning algorithms use supervised learning (SL) in which input data is labeled with the expected output. The algorithm, or network, is repeated trained with the input data until the output error is small enough.



$$y = \sigma(\boldsymbol{xw} + b)$$

Figure 5: *Basic node used in most machine learning architectures*

The basic unit of machine learning is a node as shown in Figure 5. The canonical FFNN model consists of an input layer, a hidden layer and an output layer. Each layer is constructed from interlinked nodes that generates a value (usually between -1 and 1 or 0 and 1). The individual node model is shown in Figure 5.

The node is based on the biological neuron, where dendrites bring in sensory information in the form of bioelectric impulses until the neuron activates and sends another signal through its outputs. The machine learning node is similar to an individual neuron in that it also sums the weighted inputs of the previous layer, sometimes with a bias, and transforms the combined sum with a non-linear activation function, σ before producing an output that becomes the input to other nodes or an output itself. The node equation is given by

$$y = \sigma(wx + b) \tag{8}$$

where w is an array of weights for the connections between the previous layer and the current layer, x is a vector of input values from the previous layer, and b is an optional bias value. Common activation functions include the sigmoid, tanh, and relu functions. A general property for activation functions is that they normalize the output and have a continuous first order derivative that can be used during the training process [**?**].

When many, or thousands, of nodes are used in a machine learning architecture, they become an artificial neural network (ANN). Because of the complex interconnections and nonlinear activation functions, ANNs have been successfully used to approximate complex functions and are often called 'universal approximators' [**? ?**]. The basic ANN is a feed-forward neural network (FFNN) as shown in Figure 6. It includes an input layer that takes the input data features and distributes it to hidden layers for processing. The hidden layers due the bulk of the ANN calculations because of the interconnections between nodes. Each interconnection has a weight that can be updated or turned off. An output layer converts the final calculations into a binary category or a continuous value that may require further re-mapping.

The benefits of using ANNs also include not requiring *a priori* assumptions of the data used

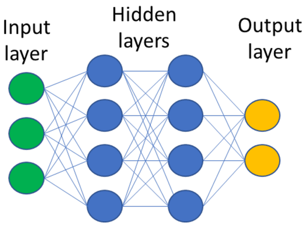


Figure 6: *Feed-forward neural network architecture*

for training and not requiring weighting of initial inputs [**?**]. In practice, dimensionality reduction is often used to remove inputs to the model that are not independent and identically distributed (IID) or offer little influence to the overall training.

Training of ANNs use gradient-based optimization to update the weights that interconnect the nodes. Through a series of back propagation, the weights are individually modified in an iterative process. The training data is then run through the network again in order to measure the error again. Each complete cycle of training is called an epoch. There is no ”one-size-fits-all” architecture and a key challenge of using machine learning tools is selecting an appropriate architecture based on the datasets available and the desired output [**?**].

This research uses datasets generated from RANS-CFD simulations on 61 different hull designs under various conditions defined by Froude number (Fn), leeway (β), and heel angle (ϕ) to train multi-layer neural networks in order to interpolate component forces under scenarios outside of the training set provided.

iii. Training set: Delft Wind Assist Series

A total set of 1,567 different RANS-CFD runs were prepared and executed over the period of 2016 and 2019 using the Numeca ISIS-CFD package. Vessel sailing performance for a range of Froude number (Fn), leeway (β), and heel angle (ϕ). The parameter space for hull geometry is indicated in Figure 7 and Table 2.

T/L Draft to lenth ratio

C_P Prismatic coefficient $\frac{A_{\text{Midship}}}{LBT}$

C_M Midship coefficient $\frac{A_{\text{Midship}}}{BT}$

Table 1: *Simulation matrix for* DWA *hulls*

Parameter	**Values**
Fn	0.126, 0.168, 0.21
Leeway (β)	[0°-9°]
Heel (ϕ)	0°, 10°, 20°

The Delft Wind Assist series is a set of 60 hull forms developed by researchers at Delft University of Technology. The hull forms are in a systematic way so that the influence of significant form coefficients for sailing behavior may be isolated and studied. The series is set up to span a design space that is presently meaningful for the application of wind-assist propulsion, summarized in Table 2.

The DWA is composed of three sub-series's:

1st Series	Variations on the Ecoliner concept [**?**]
2nd Series	19th century clipper ships
3rd Series	Low-C_P ships (ferry, cruise, and ro-ro types)

A multi-layer feed-forward neural network was prepared in Python 3.7 using the Keras library. Hyper parameters of the FFNN are shown in Table 3.

The number of hidden nodes was 6 times the input vector size (19 input features) in order to manage the complexity of the CFD output results. A low dropout (1%) was used instead of 10-20% to allow more training. Dropout is a technique that addresses overfitting, or over learning the training data, by randomly removing connections from the network forcing the training to generalize the results [**?**]. By reducing the dropout rate, we allowed the network to overfit under the assumption that the input parameters correlated to linear results in the CFD model.

Input parameters were normalized by dividing each value within the parameter column with the maximum value of the column. Heel and leeway angles were converted into

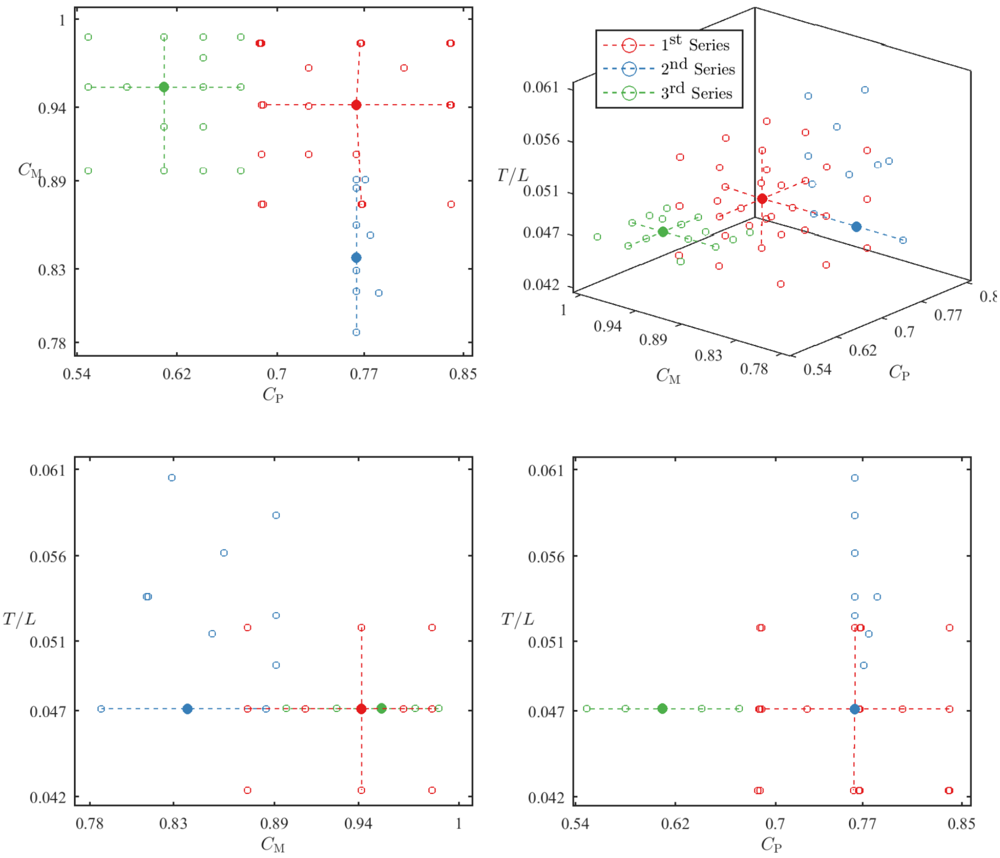


Figure 7: *Delft wind assist series, showing main parameter variations for T/L, C_P, and C_M*

cosine and sine components. Output values were multiplied by a factor of 100 to raise the small fractional values closer to 1 and then remapped using a MinMax scaler. Sequential order was not required for training so the dataset was randomly shuffled as a whole and the split into training and testing sets. A total of 85% of the values were used for training and validating with 15% reserved to test the trained network.

III. Results

The model was trained for 200 epochs in which all the training data was fed to the network in batches of 8 observations before the weights were allowed to update using the average error (loss) of the batch. Of the 85% of the total observations used for training, 15% was used to validate the accuracy of the model after each epoch. A summary of the data breakdown is shown in Table 4.

The network accuracy increased steadily over the first 20 epochs as shown in Figure 8. Both training and validation accuracies stayed constant with gradual improvement over the rest of the training cycle with losses oscillating around a central value after 100 epochs. The overall training accuracy of the network was calculated as 94.9%.

Applying the reserved test data to the trained model showed some error as measured using Mean Absolute Percent Error (MAPE) given as

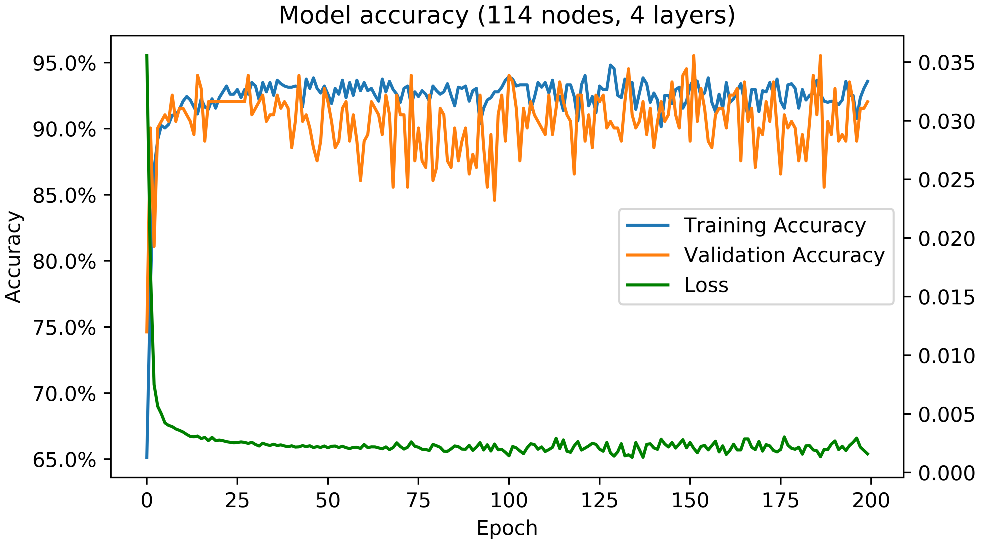


Figure 8: *Training accuracies of FFNN after 200 epochs showing validation accuracy and loss.*

$$M = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - P_i}{A_i}\right| \qquad (9)$$

where A is the CFD modeled value and P is the network predicted value. MAPE results for each force component is shown in Table 5.

Table 2: *Hydrostatics for hulls of the Delft Wind Assist Series (features of the ANN model)*

	C_B	C_P	C_M	C_{WP}	L/B	B/T	T/L	Deadrise
DWA (N=60)								
series max	0.827	0.840	0.988	0.925	8.44	3.54	0.061	0°
series min	0.493	0.549	0.787	0.747	6.00	2.16	0.042	0°
1st Series (N=33)								
Hull 1 (Parent)	0.719	0.764	0.942	0.883	7.67	2.77	0.047	0°
series max	0.827	0.840	0.984	0.925	8.44	3.42	0.052	0°
series min	0.601	0.686	0.874	0.832	6.90	2.29	0.042	0°
2nd Series (N=11)								
Hull 34 (Parent)	0.641	0.764	0.838	0.883	7.67	2.77	0.047	10°
series max	0.687	0.782	0.891	0.890	7.67	2.77	0.061	14°
series min	0.602	0.764	0.787	0.874	7.67	2.16	0.047	6°
3rd Series (N=15)								
Hull 45 (Parent)	0.582	0.610	0.954	0.805	6.00	3.54	0.047	0°
series max	0.663	0.671	0.988	0.841	6.00	3.54	0.047	0°
series min	0.493	0.549	0.897	0.747	6.00	3.54	0.047	0°

Table 3: *Network hyperparameters used to build the feed forward neural network*

Neural Network Parameter	Value
Hidden layers	2
Hidden layer nodes	114
Input and hidden layer activation function	relu
Output activation function	tanh
learning rate (α_{lr})	0.002
Optimizer	Nesterov Adam (Nadam)
Loss Function	Mean Square Error
Dropout	0.01
Batches	8

Table 4: *Breakdown of data used for network training*

Reserved Task	Observations
Training data	1136
Validation data	201
Testing data	237

The network error was considered to be within the CFD generated error.

i. Test Case - Hull 1

One of the objectives of the applying a machine learning solution to CFD generated data is to provide finer resolution for operational runs in between CFD runs. Each CFD model run can be assumed to take 4-6 hrs of computational time depending on the model, complexity and processing power. As a test case, the resolution between runs used to generate the initial training set for Hull 1 was expanded. The initial scenarios of 63 cases was increased to 630 by increasing the leeway and heel angles. The hull parameters and froud numbers were left unchanged.

The input observations were generated and processed using the same maximum column values used for the original training data and applied to the pre-trained network. The results were generated almost instantly as compared to waiting 105 days to run the 630 models individually. The average error for the different runs was 13.2% with the individual components shown in Table 6.

Table 5: *MAPE of the trained network using reserved test data.*

Component	Xn	Yn	Nn	Total
RANS-CFD	7.7%	9.9%	7.5%	
MAPE	12.9%	17.2%	42.2%	24.1%

Table 6: *MAPE results applying Hull 1 parameters to the trained network*

Component	Xn	Yn	Nn	Total
MAPE	17.5%	15.5%	6.7%	13.2%

The differing errors of the individual hull compared to the average error for all 60 different hulls in Table 5 suggests that different hulls have higher error rates than other. This must be investigated in order to identify how the network accommodates variation within hull design

IV. Discussion

A feed forward neural network was

V. AUTHORS BIOGRAPHY

Brief biographies are required for all authors e.g.:

B. Freemanis a Senior Project Manager at Lakes Software. He is responsible for researching new machine learning applications and conducting complex air dispersion modeling using CFD methods. He holds a Ph.D. in Environmental Engineering from the University of Guelph, Canada. His current interests includes deep learning analysis of spatio-temporal events and multi-sensor integration.

N. van der Kolk holds the current position of [job title] at [name of organisation]. He is responsible for [brief job description]. His previous experience includes [previous experience relevant to the paper], etc.

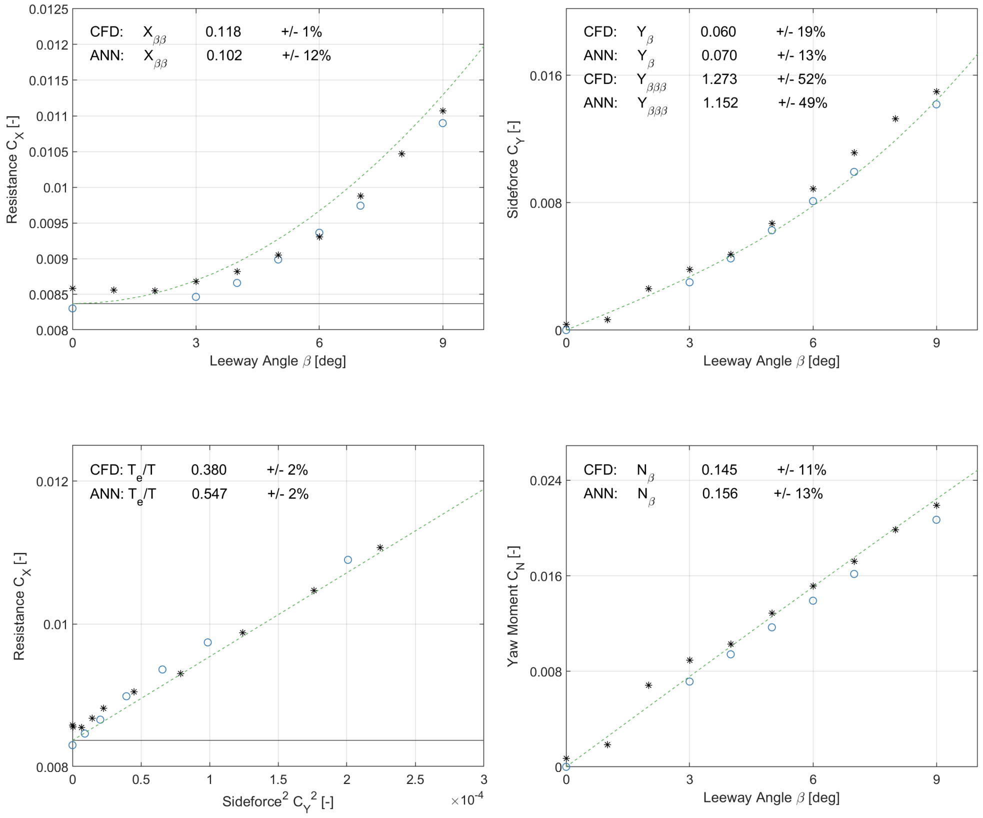


Figure 9: *Simulation results showing N-N model predictions for* DWA *Parent hull.*