MACHINE LEARNING BASED HYDRO-MECHANIC MODELING FOR SAILING COMMERCIAL SHIPS

N. van der Kolk, Blue Wasp, Netherlands, nvanderkolk@bluewaspmarine.com.
B. Freeman, Lakes Software, Canada, brian.freeman@weblake.com.

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. To expedite results under different hull designs, or in the context of scenario analysis of the global shipping 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% accuracy–commensurate with RANS-CFD uncertainty– allowing for rapid, accurate reproduction of wind-assist vessel response.

Nomenclature

q	Dynamic pressure used for non-dimensionalisation
A_{Lat}	Lateral area Lt
$A_{\mathrm{WP}}/S_{\mathrm{Wet}}$	Waterplane area to wetted surface area ratio
B/T	Beam to draft ratio
C_{B}	Block coefficient $\frac{\nabla}{LBT}$
CLR	Center of lateral resistance [m]
C_{M}	Midship coefficient $\frac{A_{\mathrm{Midship}}}{BT}$
C_{N}	Non-dimensional coefficient for yaw
C_{P}	Prismatic coefficient $\frac{A_{\mathrm{Midship}}}{LBT}$
C_{WP}	Waterplane coefficient $\frac{A_{\mathrm{WP}}}{LB}$
C_{Y}	Non-dimensional coefficient for sideforce (Sway)
$\Delta R_{\mathrm{Sailing}}$	Change in vessel resistance due to sailing condition [kN]
Fn	Froude Number
X	Flow-aligned component of vessel body force [kN]
Y	Flow-normal component of vessel body force[kN]
$\frac{h}{R_{\mathrm{b}}}$	Keel height to bilge radius ratio
L/B	Length to beam ratio
$\frac{L}{\nabla^{\frac{1}{3}}}$	Length to displacement ratio
LOA	Length over all
N	Vessel yawing moment [kNm]
N_{β}	Linear maneuvring coefficient for yaw at leeway
$N_{\beta\beta\beta}$	Non-linear maneuvring coefficient for yaw at leeway (third-order)
P_{WASP}	Available wind-assist power [kW]
$\frac{R_{\mathrm{b}}}{T}$	Bilge radius to draft ratio
$\frac{SA}{\nabla^{\frac{2}{3}}}$	Sail area to displacement ratio
$\frac{T_{\mathrm{e}}}{T}$	Vessel Effective draft (non-dimensionalised with draft T)
T/L	Draft to lenth ratio
TWA	True wind angle (vessel reference frame) [degree]
TWD	True wind direction (Earth reference frame) [degree]
TWS	True wind speed [knots]
V_{ref}	Vessel reference speed [kts]
$X_{\beta\beta}$	Maneuvring coefficient for resistance at leeway (second-order)
X_{ϕ}	Resistance due to vessel heel [kN]
T_{Aero}	Aerodynamic thrust generated by the WASP propulsor [kN]
X_{Tot}	Total resistance (including windage) [kN]
Y_{β}	Linear maneuvring coefficient for sideforce at leeway
$Y_{\beta\beta\beta}$	Non-linear maneuvring coefficient for sideforce at leeway (third-order)
Y_n	Sectional sideforce [kN/m]

1 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 [5, 16, 25, 4]. This promise of substantial reductions in emissions, for both local pollutants and for greenhouse gases, is achievable in the near-term.

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 [23, 4, 28]. These methods require intensive computational resources and complex software to generate results, thereby 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 for the global fleet, 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 transport efficiency.

Varying levels of vessel modeling may be appropriate at different stages in the analysis of wind-assist. For power-spectrum based scenario analysis considered above, a one degree-of-freedom model is often used. In this case, vessel operating constraints are typically not considered due to computational constraints. Also in the design process, a optimization routine will often need to evaluate several candidates, precluding elaborate RANS-CFD simulation or towing tank testing. In these cases, the in-built force models may be used, as developed in this paper.

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 over the chosen degrees of freedom. The performance of the ship is represented by several metrics, for example, the available wind-assist power, P_{WASP}, is plotted in fig. 1. P_{WASP} considers the wind-assist propulsive power against any power needed to operate the wind propulsor. It is defined as:

$$P_{\mathrm{WASP}} = \frac{V_{\mathrm{ref}}}{\eta_{\mathrm{T}}} \left(T_{\mathrm{Aero}} - \mathbf{\Delta R}_{\mathrm{Sailing}}\right) - \frac{P_{\mathrm{Rotor}}}{\eta_{\mathrm{Gen}}} \qquad (1)$$

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.

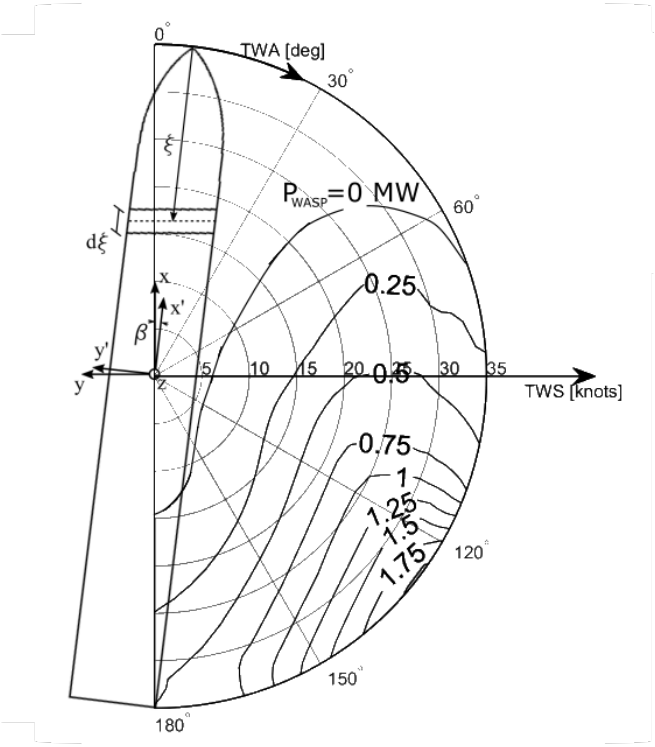


Figure 1: Available wind-assist power, P_{WASP}. Vessel coordinate system shown.

2 METHODOLOGY

2.1 HYDRO-MECHANIC RESPONSE: $\mathbf{\Delta R}_{\mathrm{Sailing}}$

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 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 fig. 1.

The ship adopts a heel and leeway angle to support the sailplan. 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 [14]. 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.

A second quantity of principal interest for this modeling for wind-assist vessels is the center of effort for the distribution

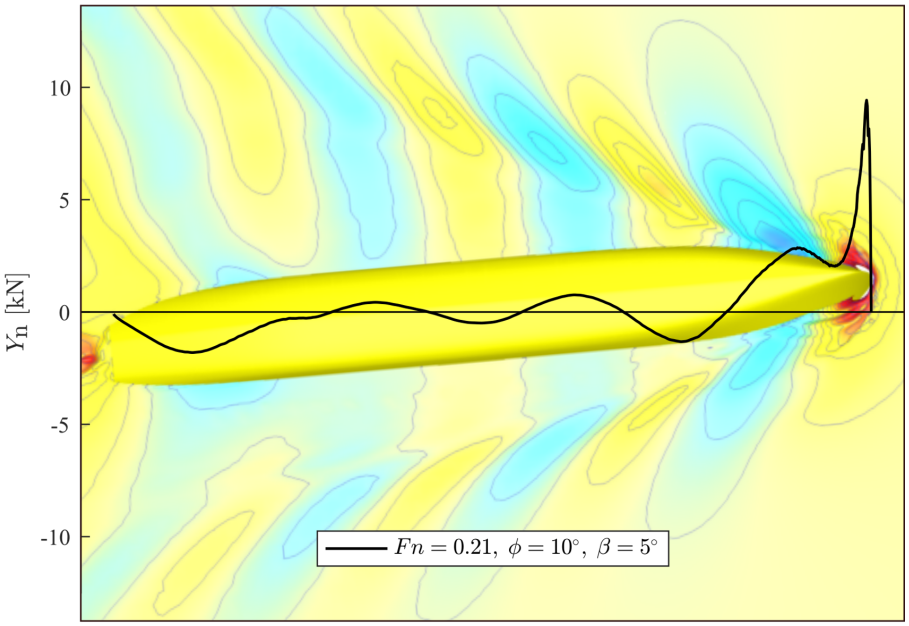


Figure 2: Distribution of hydrodynamic sideforce, showing wave system (viewpoint is below the ship). Hull # 1, leeway angle $\beta = 5°$, heel angle $\phi = 10°$. As the vessel heels, the fore and aft shoulders are brought close to the free surface, causing flow constriction and acceleration. This effect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

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.

$$CLR = \frac{C_{\mathrm{N}}}{C_{\mathrm{Y}}} \tag{2}$$

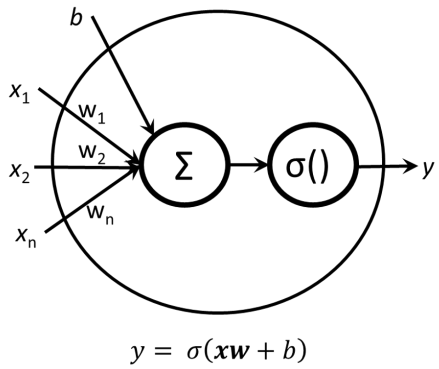
The estimation of the hydrodynamic derivatives follows from analysis of results from the database of full-scale simulation results, as described in [28, 27], for the DWA. This approach to vessel modeling follows the same methodology as [26, 11, 10, 12, 24], in the manoeuvring and sailing fields.

A full scale simulation method is adopted to resolve essential Reynolds scaling difficulties, as described in [27]. 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. 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. Complete documentation of the ISIS-CFD (FINE/Marine) solver is available [17, 1, 2, 18, 3].

2.2 UNIVERSAL APPROXIMATION WITH MACHINE LEARNING

Machine learning is a generic term that covers a broad range of analytical processes including linear regression. One of the key uses of machine learning based models is to provide universal approximators for continuous functions. The universal approximation theorem states that a feed-forward network with hidden layers can approximate any continuous functions on compact subsets of real values, R. Hornik et al (1989) first suggested that multilayer feed forward networks could meet these requirements [9]. Since then many researchers have used different configurations and activation functions to reproduce complex continuous algorithms [13, 8, 19]. 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 3: Basic node used in most machine learning architectures

The basic unit of machine learning is a node as shown in Figure 3. 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 3.
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

$$y = \sigma(wx + b) \tag{3}$$

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 [7]. 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' [20, 21]. The basic ANN is a feed-forward neural network (FFNN) as shown in Figure 4. 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.

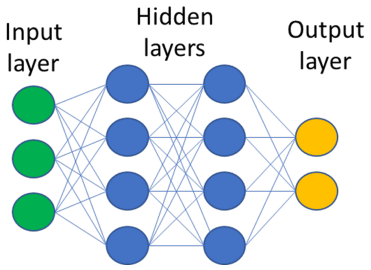


Figure 4: Feed-forward neural network architecture

The benefits of using ANNs also include not requiring *a priori* assumptions of the data used for training and not requiring weighting of initial inputs [6]. 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.

2.3 TRAINING SET: DELFT WIND ASSIST SERIES

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, including short-sea freighters, low C_{P} passenger ferries, and full-bodied tankers.

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

1st Series	General cargo ship: Variations on the Ecoliner concept [15]
2nd Series	Deadrise series, sailing ships
3rd Series	Low-C_{P} ships (ferry, cruise, and ro-ro types)

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 table 4.

The nominal testing matrix for the hulls of the Delft Wind-Assist series was 25 simulations. Simulation time for a complete drift sweep was 100 hours (192,000 CPU hours). Whereas variation for the range of speeds and leeway angles is accomplished with the drift sweep, simulation for heeled conditions required a new mesh for each heel angle.

Table 1: Setup of 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
Partition of data set	
Data Set	**Observations**
Training data	1,136
Validation data	201
Testing data	237
Total	1,574

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 1. 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 [22]. 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. Because of the relatively small training size, a small batch size was used to allow the network to update itself faster.

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 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.

3 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 1.

The network accuracy increased steadily over the first 20 epochs as shown in Figure 5. 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%.

Training set error is commensurate with DWA dataset uncertainty levels. Values for validation for the parent hull #1 are provided in table 3. RANS-CFD validation levels are adopted from [28, 27], where the verification and validation exercise are described. The propagation of errors is not considered at this stage (see Further work).

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

$$M = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - P_i}{A_i}\right| \tag{4}$$

where A is the CFD modeled value and P is the network predicted value. MAPE results for each force component is shown in Table 2. The network error was considered to be within the CFD generated error.

Table 2: MAPE of the trained network using reserved test data.

Component	**Xn**	**Yn**	**Nn**	**Total**
MAPE	12.9%	17.2%	42.2%	24.1%

The differing errors of the individual hull compared to the average error for all 60 different hulls in Table 2 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

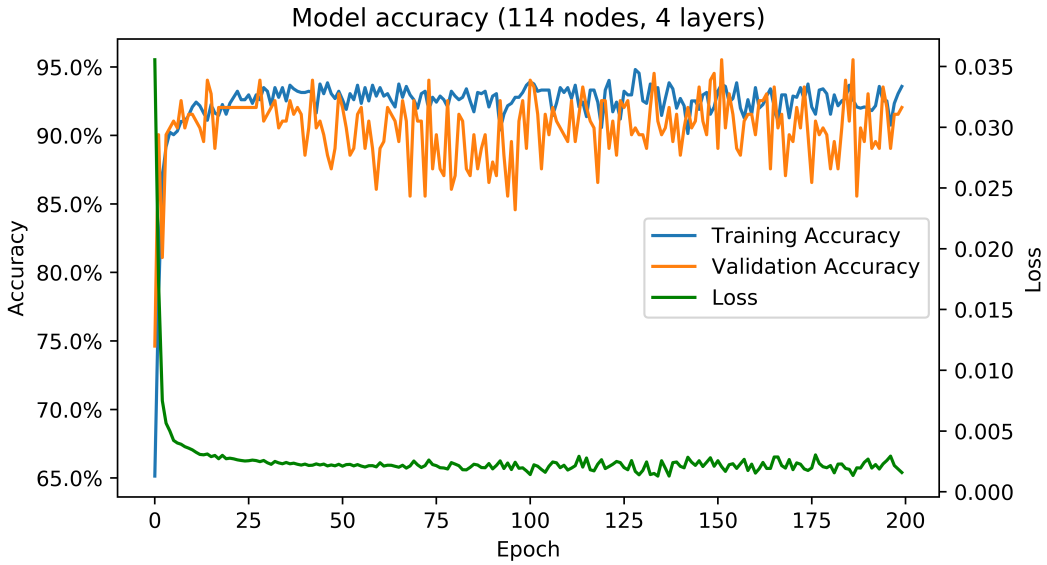


Figure 5: Training accuracies of FFNN after 200 epochs showing validation accuracy and loss.

3.1 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 Fn numbers were left unchanged.

Table 3: Test case results.

Component	RANS-CFD	ANN	E
U_x	7.7%	17.5%	
$X_{\beta\beta}$			-14%
U_y	9.9%	15.5%	
Y_β			17%
$Y_{\beta\beta\beta}$			-10%
U_n	7.5%	6.7%	
N_β			7.5

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 1,630 models individually. The average error for the different runs was 13.2% with the individual components shown in Table 3. The comparison error E for ship manuevring coefficients is is generally of the same order as the uncertainties of both the RANSE-CFD method and the ANN result.

Forces and moments are non-dimensionalized using the dynamic pressure $q = \frac{1}{2}\rho U^2$ following the maneuvering convention. A linear and third order term are assumed for β, and a linear term for ϕ.

4 CONCLUSION

A feed forward neural network was trained as a universal approximator and tested using input parameters from 61 different hulls and outputs from 1,574 RANS-CFD simulations. The network rapidly trained and experienced very high training accuracies underscoring the overall continuous nature of the RANS-CFD algorithms. The network accuracies were smaller, but still acceptable when tested with reserved datasets not used for training.

The differing errors of the individual hull compared to the average error for all 60 different hulls in Table 2 suggests that different hulls have higher error rates than other. It is also consistent with universal approximation theory that states that algorithms would be estimated best within a small subset as compared to a larger population. Working with individual hulls may produce more accurate results within the hull parameters as compared to generalizing may hulls as this research did. This must be investigated in order to identify how the network accommodates variation within hull design and extrapolate to hull designs not included in the original training set.

The time and computational saving for additional runs should not be underestimated. The process of generating the original 1,574 runs from existing RANS-CFD models that provided

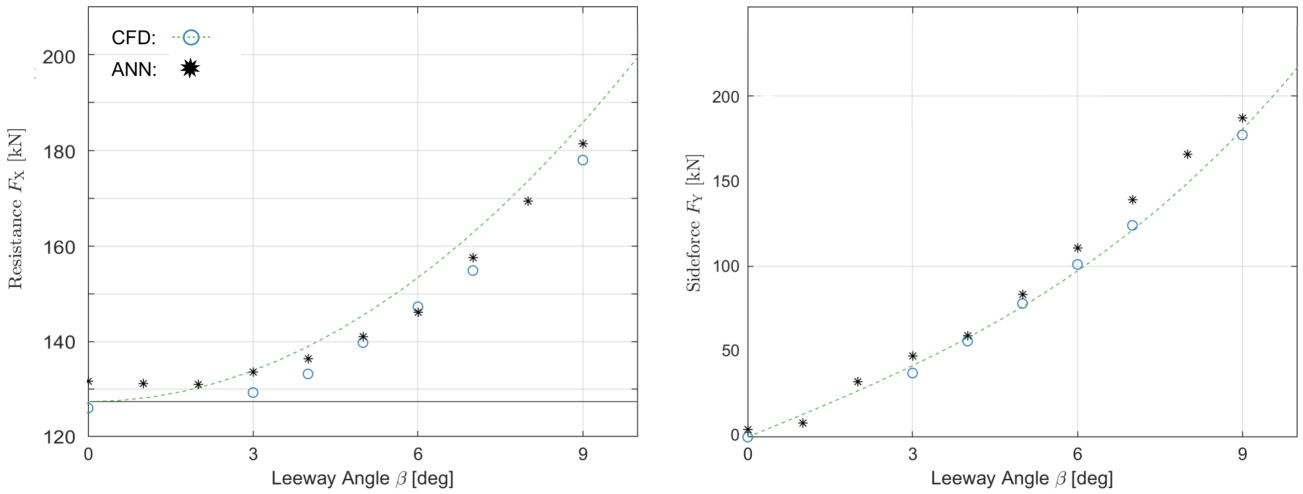


Figure 6: Simulation results showing N-N model predictions for DWA Parent hull. Fn =0.168, ϕ=0°.

the necessary training data represent 2.5M CPU hours, and took almost 2 years to accomplish. The results for 600 runs from a trained network took less than a millisecond after taking less than 2 minutes to train. Future research in this area will involve training networks of individual hulls with limited (small number of observations) training sets and estimating other output parameters of the RANS-CFD model other than force components on the hull. Further sub-series' of the DWA are also in developement: Panamax and Cape-sized (Low-T/L Series). Other opportunities include reconstructing inflow propeller wake fields, inflow velocity at rudders/ other appendages. Evaluating the body forces directly from the ANN, rather than compute standard maneuvering models, is a possible next step.

REFERENCES

[1] G. B. Deng, E. Guilmineau, P. Queutey, and M. Visonneau. Ship flow simulations with the ISIS-CFD code. CFD Workshop Tokyo, CFD Workship Tokyo, 2005.

[2] G. B. Deng, P. Queutey, and M. Visonneau. A code verificaion exercise for the unstructured finite volume cfd solver ISIS-CFD. ECCOMAS-CFD, 2006.

[3] R. Duvigneau and M. Visonneau. On the role played by turbulence closures in hull shape optimization at model and full scale. *Journal of Marine Science and Technology*, 8:11–25, 2003.

[4] R. Eggers. Operational performance of wind assisted ships. In *10th Symposium on High- Performance Marine Vehicles (HIPER)*, Cortona, Italy, 2016.

[5] T. Fujiwara, G. Hearn, F. Kitamura, M. Ueno, and Y. Minami. Steady sailing performance of a hybrid-sail assisted bulk carrier. *Journal of Marine Science and Technology*, pages 131–146, 2005.

[6] M. W. Gardner and S. R. Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14–15):2627–2636, 1998.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. Accessed 29 Jan 2018.

[8] Boris Hanin and Mark Sellke. Approximating continuous functions by relu nets of minimal width. *ArXiv*, abs/1710.11278, 2017.

[9] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989.

[10] S. Inoue, H. Masayoshi, and K. Katsuro. Hydrodynamic derivatives and ship manoeuvering. *International Shipbuilding Progress*, 1981.

[11] W. Jacobs. Estimation of stability derivatives and indices of various ship forms, and comparison with experimental results. *Journal of Ship Research*, pages 135–163, 1966.

[12] J. A. Keuning and U. B. Sonnenberg. Approximation of the hydrodynamic forces on a sailing yacht based on the 'delft systematic yacht hull series'. In *HISWA Symposiumon Yacht Design and Yacht Construction*, pages 99–152, 1998.

[13] Hrushikesh Mhaskar and Tomaso A. Poggio. Deep vs. shallow networks : An approximation theory perspective. *ArXiv*, abs/1608.03287, 2016.

[14] M. M. Munk. Remarks on the pressure distribution over the surface of an ellipsoid moving translationally through a perfect fluid. Technical report, NACA, 1924.

[15] Dykstra NA. Ecoliner, 2010.

[16] P. Naaijen, W. Shi, and J. Kherian. Assessing the fuel savings by using auxiliary wind propulsion from traction kites. *RINA - Ship Design and Operation for Environmental Sustainability - Papers*, pages 27–37, 2010.

[17] Numeca Inc. ISIS theoretical manual. Technical report, Numeca International.

[18] P. Queutey and M. Visonneau. An interface capturing method for free-surface hydrodynamic flows. *Computers and Fluids*, 36:1481–1510, 2007.

[19] Nimrod Segol and Yaron Lipman. On universal equivariant set networks. *ArXiv*, abs/1910.02421, 2019.

[20] Amel Sifaoui, Afef Abdelkrim, and Mohamed Benrejeb. On the use of neural network as a universal approximator. *International Journal of Sciences*, 2:386–399, 08 2008.

[21] Sho Sonoda and Noboru Murata. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43(2):233 – 268, 2017.

[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[23] Tahsin Tezdogan, Yigit Kemal Demirel, Paula Kellett, Mahdi Khorasanchi, Atilla Incecik, and Osman Turan. Full-scale unsteady rans cfd simulations of ship behaviour and performance in head seas due to slow steaming. *Ocean Engineering*, 97:186 – 206, 2015.

[24] S. L. Toxopeus. Practical application of viscous-flow calculations for the simulation of manoeuvering ships. *PhD Thesis*, 2011.

[25] Michael Traut, Paul Gilbert, Conor Walsh, Alice Bows, Antonio Filippone, Peter Stansby, and Ruth Wood. Propulsive power contribution of a kite and a Flettner rotor on selected shipping routes. *Applied Energy*, 113:362–372, 2014.

[26] S. Tsakonas. Effect of appendage and hull form on hydrodynamic coefficients of surface ships. Technical report, Hoboken, NJ, 1959.

[27] N.J. van der Kolk, I. Akkerman, J. A. Keuning, and R.H.M. HUuijsmans. Part 2: Simulation methodology and numerical uncertainty for rans-cfd for the hydrodynamics of wind-assisted ships operating at leeway angles. Manuscript accepted for publication.

[28] van der Kolk N. J., J. A. Keuning, and R. H. M. Huijsmans. Part 1: Experimental validation of a RANS-CFD methodology for the hydrodynamics of wind-assisted ships operating at leeway angles. *Ocean Engineering*, 178:375–387, 2018.

5 AUTHORS BIOGRAPHY

N. van der Kolk is co-founder of Blue Wasp, a company he started with a fellow Ph.D. student, where he is actively engaged with supporting and promoting the broader uptake of wind-assist for commercial shipping. Besides, he is presently completing his doctoral research at Delft University of Technology, entitled: *Sailing Efficiency and Course Keeping of Wind-Assisted Ships*. His interests include energy policy and sailboat racing.

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.

Table 4: 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°
1^{st} 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°
2^{nd} 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°
3^{rd} 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°