

## Article

# On the Use of Artificial Intelligence to Define Tank Transfer Functions

Pál Schmitt <sup>1,\*</sup> , Charles Gillan <sup>2</sup>  and Ciaran Finnegan <sup>2</sup>

<sup>1</sup> Queen's University Marine Laboratory

12-13 The Strand, Portaferry

Co. Down, Northern Ireland, BT22 1PF, United Kingdom; p.schmitt@qub.ac.uk

<sup>2</sup> School of Electronics, Electrical Engineering and

Computer Science, Queen's University Belfast, 16A Malone Road,

Belfast, Northern Ireland BT9 5BN, United Kingdom ;c.gillan@qub.ac.uk (C.G.); cfinnegan12@qub.ac.uk (C.F.)

\* Correspondence: p.schmitt@qub.ac.uk; Tel.: +44 7526698884

**Abstract:** Experimental test facilities are generally characterised using linear transfer functions to relate the wavemaker forcing amplitude to wave elevation at a probe located in the wavetank. Second and third order correction methods are becoming available but are limited to certain ranges of waves in their applicability. Artificial intelligence has been shown to be a suitable tool to find even highly nonlinear functional relationships. This paper reports on a numerical wavetank implemented using the OpenFOAM software package which is characterised using artificial intelligence. The aim of the research is to train neural networks to represent non-linear transfer functions mapping a desired surface-elevation time-trace at a probe to the wavemaker input required to create it. These first results already demonstrate the viability of the approach and the suitability of a single setup to find solutions over a wide range of sea states and wave characteristics.

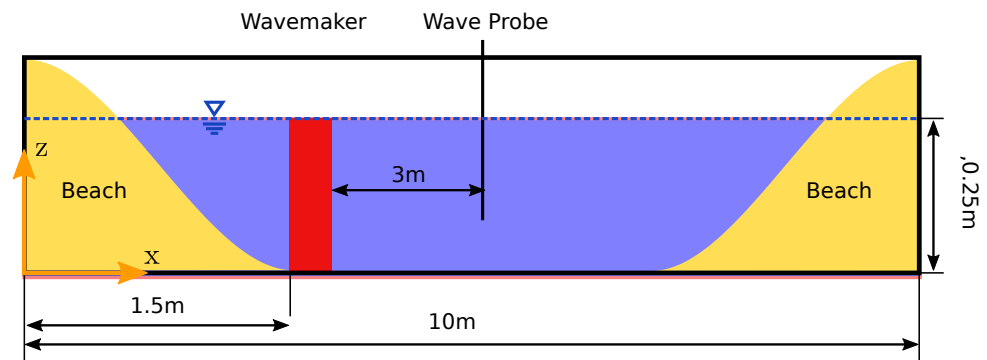
**Keywords:** tank transfer function; neural networks; machine learning; OpenFOAM; computational fluid dynamics;

## 1. INTRODUCTION

Waves are of particular interest in marine research as they play a key role in understanding fluid-structure interactions. Wave tank test facilities are used to generate desired water waves in order to model certain sea-states in a controlled environment. Both physical and simulated wavetanks exist, each with their own merits. However, in all cases wavemakers are required and some method to control wavemaker action to obtain a desired sea state or time trace of surface elevation in the tank. Early work employed analytical solutions relating piston or paddle motion amplitude to surface elevation [1]. Industry standard is now the characterisation of a wavetank using tank transfer functions (TTF) [2], describing the ratio of surface elevation to wavemaker amplitude and phase for each frequency component. Even for a flat bathymetry, non-linear extensions are often required for anything beyond very small amplitude waves to take into account bound waves and wave-wave interaction. Significant progress was achieved by extension to second order [3], but despite further progress these analytical solutions are still limited in their range of applicability and no single method encompassing all features relevant in applied ocean and coastal research exists [4–6]. Recently, large collaborative projects in offshore wind [7] and wave power [8] independently identified a need for better accuracy in the reproduction of a given surface elevation time trace in numerical simulations.

An artificial neural network (ANN) is a mathematical structure represented as a set of interconnected nodes, with the connections called neurons. Different types of neural network can be distinguished by the way in which the nodes are connected.

These nodes are in fact numeric values and the connections are multiply-accumulate operations (MAC) which are executed in sequence, corresponding to linking the neurons together. The result of each operation is used as input to an activation function which decides whether the result is fed forward to the next stage, or not, and different choices exists for the activation function. In addition, each MAC operation has several weights which are computed during the training phase.



**Figure 1.** Geometric layout for the numerical wavetank.

Calibrating a wavemaker can be regarded as solving the wave propagation problem, and thus the Navier-Stokes Equation, backwards in time. ANNs can function as universal function approximators and have been shown to be capable of solving differential equations. In our work we apply one type of ANN, the multi-layer perceptron to solve the Navier Stokes equations in our numerical wavetank. We show that it is possible to accurately map a desired wave-elevation time-series, at a measurement probe position, to the wavemaker input required to generate this time series. In the literature there is only one known application of Neural Networks to predict wavemaker inputs based on specific wave traces for calibration. The work done by [9] experimented with the first use of Non-Linear Auto-regressive Exogenous input (NARX) Neural Networks for the calibration of waves close to the breaking limit of a numerical wavetank based on a method later published in [10]. For model training, waves were first generated with random wavemaker input. The time-trace of the surface elevation of the resultant waves were then used as model input and the given random wavemaker input was used as the desired model output. With each run more data became available and the solution converged towards the desired result. These experiments indicate that NNs are a valuable tool for wavemaker calibration and could yield better results with further experimentation. One issue in the application of NN to wave calibration identified, was the short "memory" or "vanishing gradient", which has been investigated by [11]. The influence of an item in sequence weakens as the sequences goes on. This is problematic for learning long sequences, as long patterns and sparse temporal dependencies may not be recognised.

Neural networks methods have been used by other researchers in the field. [12] uses Recurrent Neural Networks (RNNs) to predict statistical properties of water wave amplitude time traces as a function of the wavemaker parameters (blower RPM and frequency). While they only predicted statistical properties of the wave produced and not a specific time-series, this still shows the promise of RNNs in predicting water waves. A novel neural network modelling approach has been proposed by [13]. This proposal presents the idea of Physics Informed Neural Networks to solve differential equations, giving an example of wave propagation. In this method, the laws of physics are invoked to constrain calculated values to a suitable range. This method may be suitable for application to the Navier-Stokes equations.

The layout of this paper is as follows. In the next section we explain the methodology that we have implemented to perform our numerical experiments. Section 3 presents the results of our experiments including application to a non-linear case. The

paper concludes with section 4 where we summarise the findings of our work and look towards future applications of it.

## 2. METHODOLOGY

### 2.1. Function approximation

Following Friedmann[14], we may state the problem of finding the inverse of the non-linear transfer function in the following way. We have an output variable  $U_i$ , the required wavemaker function, defined as a function at a discrete set of  $n$  time points,  $t_i$ . This is dependent on the function,  $F(h_i, t_i)$ , the form of which is unknown. The function depends on the height of the probe amplitude,  $h_i$  at time  $t_i$  and represents the inverse of the tank transfer function.

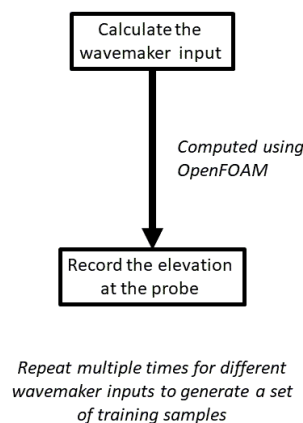
However, we have a vector  $\mathbf{v}$  which is a set of  $n$  observations  $(h_i, t_i)$   $i = 1, n$  at the probe to which we can associate a vector  $\mathbf{U}$  of  $n$  wavemaker inputs from a previous time interval. We know that these inputs and outputs are related by equation (1)

$$F(\mathbf{v}) \rightarrow \mathbf{u} \quad (1)$$

The universal approximation theorem for neural networks asserts that we can use the input and output vectors  $\mathbf{u}, \mathbf{v}$  to find an approximation for the unknown function  $F$  in equation (1) for a vector of observations,  $\mathbf{u}$ , to a unique output vector,  $\mathbf{v}$ , expressed in equation 1.

### 2.2. Generating training data

The work reported in this paper uses the numerical wavetank (NWT) shown in figure 1 for the preparation of suitable training data but the method should be applicable to any numerical and physical wave maker. The numerical wavetank employed was based on the OpenFOAM based methods described in detail in [10,15]. While a number of free surface methods are available, we chose the *interFoamsrc* volume-of-fluid solver for maximum flexibility. The wavemaker imposes vertical acceleration to the fluid, a numerical wave probe records resulting surface elevation at a distance of 3m. Beaches of 1.5m length cancel reflections, the water depth is 0.25m. We need to prepare multiple distinct instances of the input and output vectors  $\mathbf{u}, \mathbf{v}$ , defined in the previous section in order to generate sufficient data with which to train the neural network models. Figure 2 presents the workflow for this step.



**Figure 2.** Illustration of the work flow for generation of training data.

Random wavemaker input traces were generated by summing up individual contributions for each frequency  $f$  using the formula,

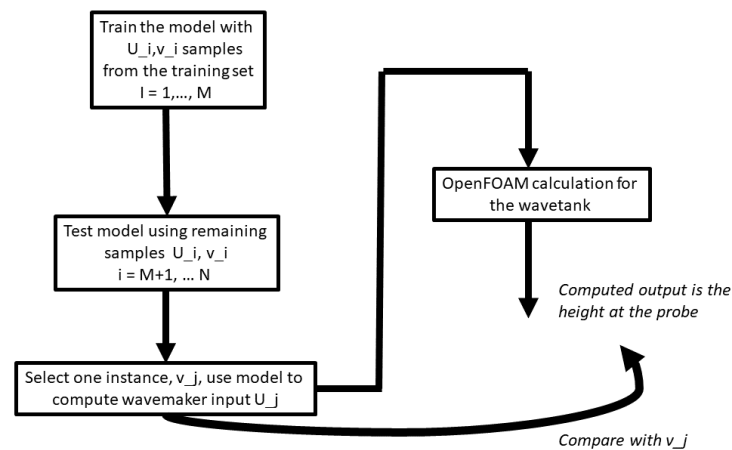
$$U_i(t) = \sum_{k=1}^{N_r} \frac{U_m}{N_r} \sin\left(\frac{2\pi}{T(k)}t - \phi(k)\right) \quad (2)$$

A Tukey filter was applied to the resulting time trace to ensure wavemaker input began with zero amplitude, avoiding shock-waves in the numerical wavetank.

This process was repeated multiple times to define a set of samples. Each sample,  $u_i$  was then used as input to the OpenFOAM program for the wavetank configuration defined earlier in this paper. The computed surface elevation at the probe, for this input, was recorded as the output from OpenFOAM producing an instance  $v_i$ . The workflow in figure 2 can be repeated to produce a training set of arbitrary size.

### 2.3. Training the neural network model

We have used the multi-layer perceptron (MLP) model of neural networks for the results that we report in this paper. There are multiple variability points, known as hyper-parameters, in an MLP model. These include the number of layers used, the number of nodes per layer and the activation function used at each layer. Varying these parameters changes the performance of the model. A standard technique is to search



**Figure 3.** Training and validating the neural network models.

over multiple values of the hyper parameters to find the model which best fits the data. Model fitting for each set of hyper-parameters is independent of every other set, that is embarrassingly parallel. This process generally requires large amounts of compute time, however compared to typical fluid dynamics problems the overhead is small and can be run on the same HPC architecture.

[16] showed that the multilayer feed-forward architecture gives neural networks the potential to be universal approximators. The author showed that as long as the activation function is continuous, bounded and non-constant, then continuous mappings can be learned uniformly over compact input sets. The classical form of the universal approximation theorem for arbitrary width and bounded depth shows that a feed forward neural network is capable of approximating any well-behaved function [17]. We thus aim to replace classical TTF with trained NN, which, once trained, should yield any required surface elevation trace without further iterative calibration steps.

2.4. Validation of the model

. It is customary to use k-fold cross validation in order to remove bias in the training data. In our work we split the data samples on an 8 : 2 ratio of training to test samples. The members of each set are chosen randomly. This process is repeated multiple times and the model giving the best performance is chosen.

A further check on the accuracy of our method is illustrated in figure 3. Once a neural network model has been chosen, we can take the wavemaker input that is predicted as required to generate a given time profile at the probe and feed this as input to the numerical wavemaker. The computed time profile at the probe can be compared to that which was fed to the neural network to produce the wavemaker input.

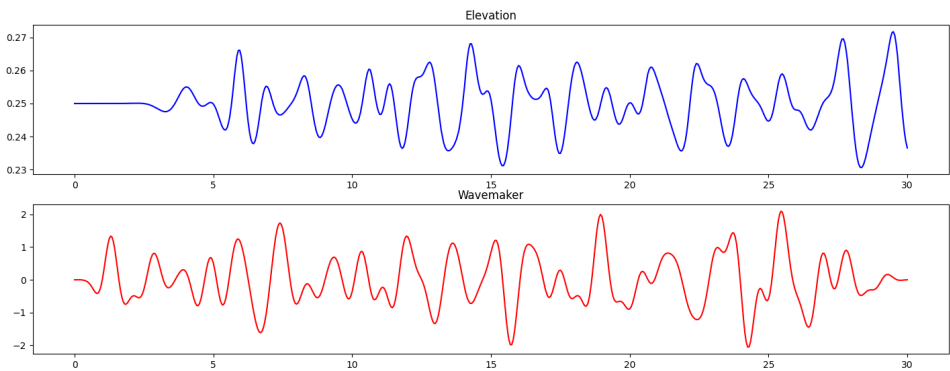
3. RESULTS

We prepared 1000 samples of wavemaker input, each of 30s duration, using equation 2) with the parameters defined in table 1. Note that we used two different values for the maximum wavemaker height,  $U_m$ . We first report on the case where  $U_m = 1$ . This corresponds to the creation of waves with relatively small heights at the wavemaker as shown on the y-axis of figure 4. In the later part of this section we report on the case of generating much larger amplitude waves using  $U_m = 50$ . Figure 4 presents an example

Table 1: Values used in equation (2) to produce the wavemaker time traces for the neural network model.

Parameter	Description	Value
$T$	Time in seconds	$[0.7 : (1.8s)]$
$N_r$	Number of realisations	1000
$U_m$	Constant	1 or 50
$t$	Time in seconds	$[0, \dots, 30]s$
$\phi(k)$	Random phase	$[0, 2\pi]$

of the input to the wavemaker and subsequent computed surface elevation of the probe. The flat line to the left of the surface elevation plot represents the time delay between



**Figure 4.** The lower figure shows the wavemaker acceleration as a function of time feeding the simulations using OpenFOAM. The upper figure shows computed surface elevation at the probe. The values on the y-axes are in metres and the x-axes in seconds.

wave generation starting at the wavemaker and the arrival of the first wave at the probe.

The most time intensive part of the computation is the search over the space of hyper parameters seeking to find a minimum in a chosen loss function. Fortunately the values taken by the hyper parameters are denumerable rather than continuous. In our work we have used the mean square error function, that is we compute the mean of squares of errors between the observations and the predictions during the training phase. It should

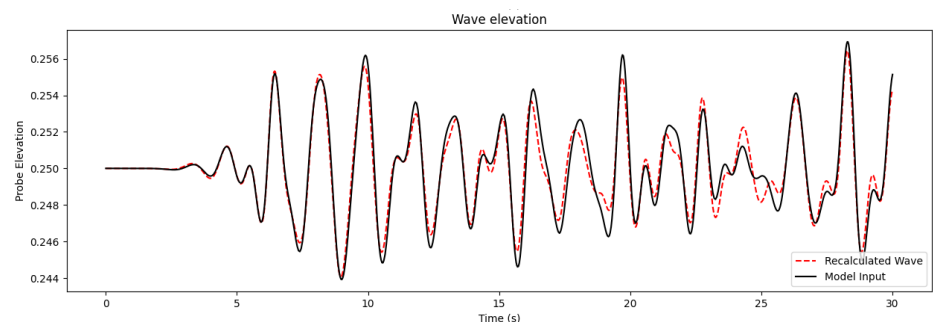
be noted though, that any other parameter or even combinations of interest could be used, like for example the height of the extremest waves or zero-crossing periods or phases of some selected components in case where resonance is more important. By construction, this search can only find a local minimum for the ANN representation. It is this property which leads to the use of large amounts of CPU time, that is to search over as large a space of hyper parameters as reasonably possible.

Table 2 shows the set of hyper parameters that we identified after an extensive search, using the mean squared error function. In the search, three activation functions were tested: Rectified Linear (ReLU), Hyperbolic Tangent (tanh) and Sigmoid. The results showed that all three activation functions reached a point of stagnation and convergence at around 350 epochs with a Mean-Squared Error of approximately 0.0033.

Table 2: Optimal values for the hyper-parameters of the neural network MLP model. These were obtained by a grid search over multiple parameters.

Parameter	Value
Training epochs	350
Learning rate	0.001
Hidden layer 1 nodes	310
Activation function	tanh
Hidden layer 2 nodes	310
Activation function	linear
Loss function	MSE
Temporal resolution	0.1s

Following the validation process discussed in the previous section and presented in figure 3, the method reproduced the probe height accurately across the range of time points. The mean squared error over a typical time trace was  $3.25 \cdot 10^{-7}$ . Figure 5 is an example comparing the two traces at the probe. After this initial demonstration of the



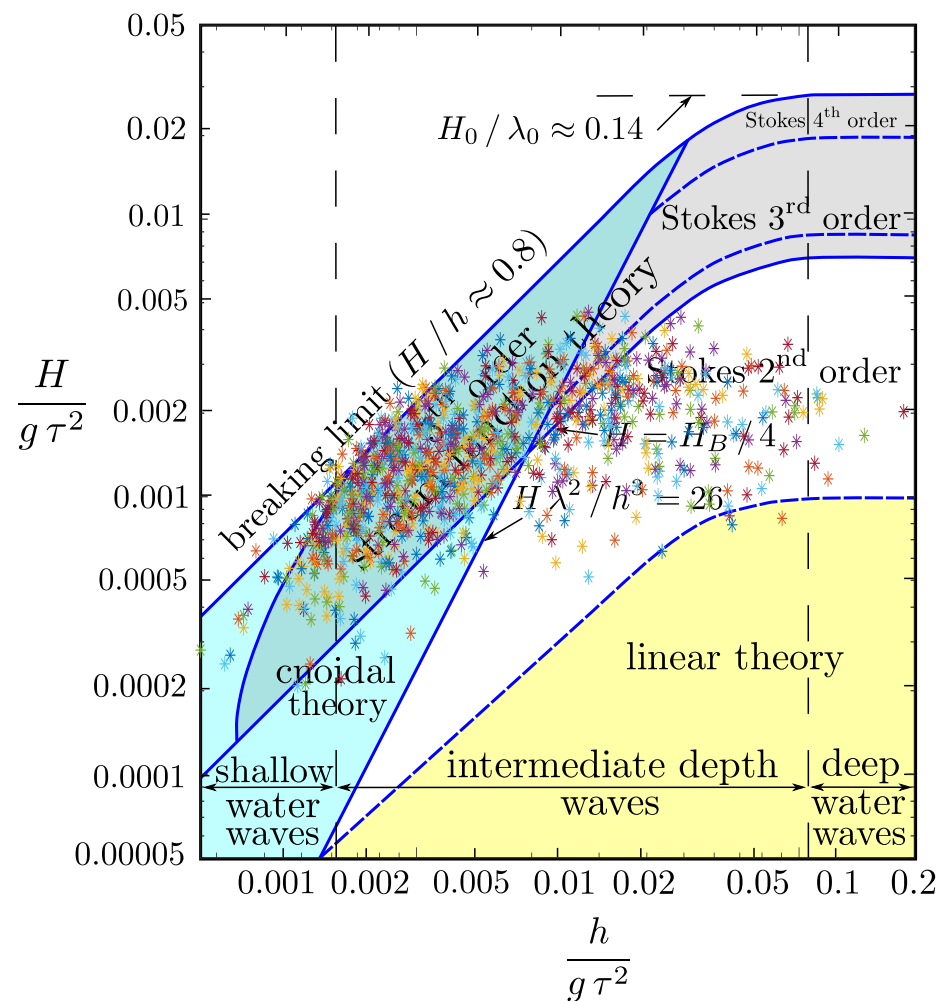
**Figure 5.** The black line is the probe elevation input to neural network model to infer the wave-maker input used to create it. When this is input to OpenFOAM, the red line is the calculated elevation of the probe.

fundamental suitability of the method, a second set of data was generated following the method described above, but with the parameter  $U_m$  set to 50. A time domain analysis of the resulting surface elevation traces was performed using custom written GNUoctave [18] code as follows:

- depth  $h$  is the mean value of the surface elevation trace
- Find zero-crossing times of surface elevation minus depth
- Iterate over every second zero-crossing time and find max and min values for given interval
- Period  $\tau$  is then the overall interval length
- wave height  $H$  is the sum of the magnitudes of maximum and minima values in each interval

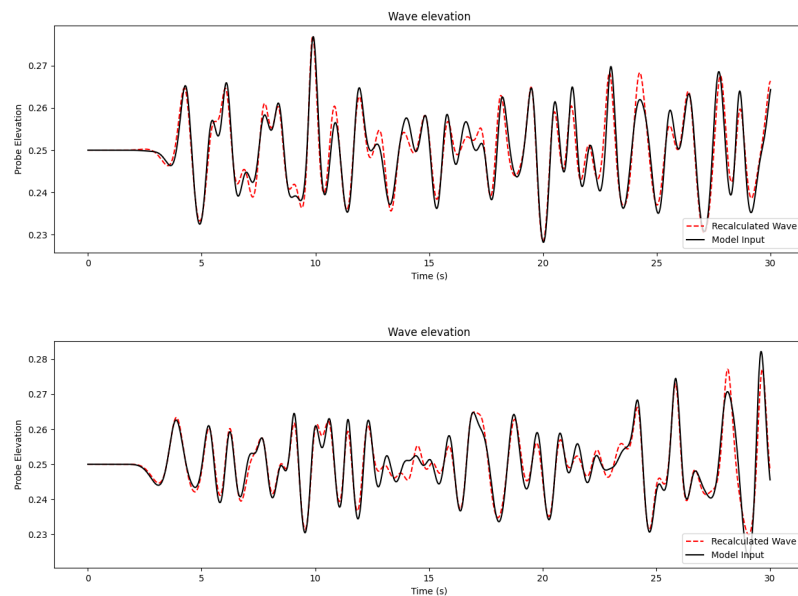
The values for each wave are non-dimensionalised with gravitational acceleration  $g$  and wave period  $\tau$  as pairs of  $\frac{H}{g\tau^2}$  and  $\frac{h}{g\tau^2}$  and plotted as a point cloud over the well known plot highlighting the validity of different wave theories [19], Fig 6. The majority of waves is highly non-linear and can best be described with 5<sup>th</sup> order stream function theory, reaching into 2<sup>nd</sup> and 3<sup>rd</sup> order stokes theory. Some few waves seem to exceed the breaking limit which requires further investigation but might well be due to the highly irregular nature of the timetrace.

The key feature to note here is, that the waves are highly nonlinear and, while direct comparison is still outstanding, could be expected to pose a formidable challenge to conventional wave calibration techniques. Figure 7 presents two example results for surface elevation using the predicted wavemaker input besides their target data. These results were achieved using the same settings found to be optimal for the small amplitude waves. Even without further optimisation agreement is good. In the first example (top) the highest peak at 10s and the lowest trough at 20s match almost perfectly in time and amplitude. In the second example (bottom), deviations for the largest waves can be observed at the end of the trace, but overall the agreement is still remarkable.



**Figure 6.** Wave parameters encountered in the larger wave cases represented as dots over the validity ranges of several theories for periodic water waves, according to [19]. Original figure adapted from Wikimedia Commons, the free media repository





**Figure 7.** Comparison between recalculated surface elevation and original data for two sets of results for the large wave setup.

### 3.1. Computational environment used

Initial investigations of suitable neural network architectures were performed using version 3.8.5 of the Python language and using the GPU enabled version of Tensorflow. This software was executed on a system which has an AMD Ryzen 7 3700x CPU and a NVIDIA GeForce RTX 2070 GPU. This system has 8GB RAM and the operating system Ubuntu 20.04.1 LTS. Production runs were performed at the EPSRC funded Tier 2 Northern Ireland HPC centre.

## 4. CONCLUSIONS

In this paper we present a method which uses neural networks to infer the wave-maker input time-series required to produce a desired wave-elevation time-series at a probe in a fixed location. We present results using a Multi-Layer Perceptron neural network which has been tuned and trained over 800 samples of random input and output data. The neural network consistently produced outputs that, when used on the wavemaker, yielded accurate replications of the desired waves. Producing an inference from the model for a single input wave was fast and efficient, showing great promise as an alternative calibration method to the iterative tuning methods based on linear tank transfer functions.

Although not demonstrated here explicitly, it should also be noted that short time traces can be used effectively. Short time traces of data can cause issues in conventional methods based on Fast Fourier Transforms, but are often the only feasible option in computationally intensive simulations which might only be used to investigate two or three wave cycles of interest.

While the results from this model are promising, this work is just a first proof of concept and foundation for further investigations.

The current model required input sequences with a strict number of time-steps, 301 steps in the case of the final trained model, and only returned output sequences with the same number of steps. Attempting a model inference/prediction for an input wave-form that has a time span longer than that of the training data will be an interesting future challenge.

Furthermore, while inferencing with the neural network method is very fast the process of generating training data and training the model can be time consuming.



However, since the required simulations can be run in parallel and often in 2 dimensional cases on HPC facilities the overall run time doesn't need to be long. In experimental facilities, where each dataset must be generated in real time and in sequence this might be more of an issue. But even then, the data generation can be run once in a highly automated way with minimal manual interaction and subsequently instantly provide virtually any surface elevation trace required by the user. The effect of fewer training samples on the model accuracy certainly warrants further investigation.

This research demonstrates the potential for the use of machine learning technology in the calibration of wavemakers, and highlights how further investigation and development could result in improvements in the utilisation of the technology.

## ACKNOWLEDGEMENTS

The Bryden Centre project funded Paí Schmitt and is supported by the European Union's INTERREG VA Programme, managed by the Special EU Programmes Body (SEUPB). Computations were performed on the Kelvin-2 HPC system based at QUB, funded by EPSRC grant EP/T022175/1.

## DISCLAIMER

The views and opinions expressed in this paper do not necessarily reflect those of the European Commission or the Special EU Programmes Body (SEUPB).

## References

1. Dean, R.G.; Dalrymple, R.A. *Water Wave Mechanics for Engineers and Scientists*; WORLD SCIENTIFIC, 1991; <https://www.worldscientific.com/doi/10.1142/1232>.
2. Masterton, S.; Swan, C. On the accurate and efficient calibration of a 3D wave basin. *Ocean Engineering* **2008**, *35*, 763–773. doi:10.1016/j.oceaneng.2008.02.002.
3. Schäffer, H. Second-order wavemaker theory for irregular waves. *Ocean Engineering* **1996**, *23*, 47–88. cited By 239, doi:10.1016/0029-8018(95)00013-B.
4. Eldrup, M.R.; Lykke Andersen, T. Applicability of Nonlinear Wavemaker Theory. *Journal of Marine Science and Engineering* **2019**, *7*, doi:10.3390/jmse7010014.
5. Khait, A.; Shemer, L. Nonlinear wave generation by a wavemaker in deep to intermediate water depth. *Ocean Engineering* **2019**, *182*, 222–234. cited By 8, doi:10.1016/j.oceaneng.2019.04.065.
6. Khait, A. Third-order generation of narrow-banded wave trains by a wavemaker. *Ocean Engineering* **2020**, *218*, 108200. doi:10.1016/j.oceaneng.2020.108200.
7. Robertson, A.; Bachynski, E.E.; Gueydon, S.; Wendt, F.; Schünemann, P. Total experimental uncertainty in hydrodynamic testing of a semisubmersible wind turbine, considering numerical propagation of systematic uncertainty. *Ocean Engineering* **2020**, *195*, 106605. doi:10.1016/j.oceaneng.2019.106605.
8. Ransley, E.; Yan, S.; Brown, S.; Hann, M.; Graham, D.; Windt, C.; Schmitt, P.; Davidson, J.; Ringwood, J.; Musiedlak, P.; Wang, J.; Wang, J.; Ma, Q.; Xie, Z.; Zhang, N.; Zheng, X.; Giorgi, G.; Chen, H.; Lin, Z.; Qian, L.; Ma, Z.; Bai, W.; Chen, Q.; Zang, J.; Ding, H.; Cheng, L.; Zheng, J.; Gu, H.; Gong, X.; Liu, Z.; Zhuang, Y.; Wan, D.; Bingham, H.; Greaves, D. A blind comparative study of focused wave interactions with floating structures (CCP-WSI blind test series 3). *International Journal of Offshore and Polar Engineering* **2020**, *30*. doi:10.17736/ijope.2020.jc774.
9. Schmitt, P. Steps Towards a Self Calibrating, Low Reflection Numerical Wave Maker Using Narx Neural Networks. Proceedings of the VII International Conference on Computational Methods in Marine Engineering, 2017. International Conference on Computational Methods in Marine Engineering ; Conference date: 15-05-2017 Through 17-05-2017.
10. Schmitt, P.; Windt, C.; Davidson, J.; Ringwood, J.V.; Whittaker, T. Beyond VoF: alternative OpenFOAM solvers for numerical wave tanks. *Journal of Ocean Engineering and Marine Energy* **2020**. doi:10.1007/s40722-020-00173-9.
11. Diaconescu, E. The Use of NARX Neural Networks to Predict Chaotic Time Series. *WSEAS Trans. Comp. Res.* **2008**, *3*, 182–191.
12. Fu, T.; Fullerton, A.; Rice, J.; Fu, T.; Hess, D. A novel approach to calibrating wavemakers and generating wavemaker transfer functions. Technical Report NSWCCD-50-TR-2007/028, Carderock Division Naval Surface Warfare Center, West Bethesda, Maryland 20817-5700, 2007. doi:10.1115/OMAE2007-29230.
13. M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* **2019**, *378*, 686–707.
14. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics* **2001**, *29*, 1189–1232.
15. Windt, C.; Davidson, J.; Schmitt, P.; Ringwood, J.V. On the Assessment of Numerical Wave Makers in CFD Simulations. *Journal of Marine Science and Engineering* **2019**, *7*. doi:10.3390/jmse7020047.

- 
16. Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks* **1991**, 4, 251–257. doi:10.1016/0893-6080(91)90009-T.
  17. Hassoun, M. *Fundamentals of Artificial Neural Networks*; MIT Press: Cambridge, Massachusetts, USA, 1995.
  18. Eaton, J.W.; Bateman, D.; Hauberg, S.; Wehbring, R. *GNU Octave version 5.2.0 manual: a high-level interactive language for numerical computations*, 2020.
  19. Méhauté, B. *An Introduction to Hydrodynamics and Water Waves*; Springer-Verlag: New York, 1976.