

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/333681969>

SHIP AS A WAVE BUOY – USING SIMULATED DATA TO TRAIN NEURAL NETWORKS FOR REAL TIME ESTIMATION OF RELATIVE WAVE DIRECTION

Conference Paper · June 2019

CITATIONS

5

READS

343

2 authors, including:



Bülent Düz

Maritime Research Institute Netherlands

33 PUBLICATIONS 204 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



PIV and CFD [View project](#)



Machine learning in maritime applications [View project](#)

SHIP AS A WAVE BUOY - USING SIMULATED DATA TO TRAIN NEURAL NETWORKS FOR REAL TIME ESTIMATION OF RELATIVE WAVE DIRECTION

Bart Mak*

Bülent Düz*

Maritime Research Institute Netherlands (MARIN)
P.O. Box 28, 6700AA, Wageningen
The Netherlands
Email: {b.mak, b.duz}@marin.nl

ABSTRACT

Being able to give real time on-board advice, without depending on extensive sets of measured data, is the ultimate goal of the digital twin concept. Ideally, the models used in a digital twin only rely on current in-service data, although they have been built using simulated and possibly some measured data. Working with just the 6-DOF motions of a ship, can the local sea state reliably be estimated using the digital twin concept? Does a general model exist to do so, without the need to measure or simulate the particular ship?

In this paper, we discuss how simulations of an advancing ship, subjected to various sea states, can be used to estimate the relative wave direction from in-service motion measurements of the corresponding ship. Various types of neural networks are used and evaluated with simulated data and measured data. In order to study the generalization power of the neural networks, a range of ships has been simulated, with varying lengths, drafts and geometries. Neural networks have been trained on selections of the ships in this extended training set and evaluated on the remaining ships.

Results show that the developed neural networks give a remarkable performance in simulation data. Furthermore, generalization over geometry is very good, opening the door to train a general model for estimating sea state characteristics. Using the same model for in-service measurements does not perform well enough yet and further research is required. The paper will in-

clude discussion on possible causes for this performance gap and some promising ideas for future work.

1 INTRODUCTION

Good quality sea state information is important for operations at sea. Being able to accurately assess the situation leads to improved personal and environmental safety and better efficiency. Because of this, in the past few decades many methods have been proposed to infer sea state information from indirect measurements. The use of some real time method to assist in specific situations is sometimes referred to as *Digital Twin*. This concept includes both physical models and data driven models.

Iseki [1] and Tannuri [2] have proposed methods that optimize the inferred sea state such that the spectrum of the expected ship motions matches the actual motions. This match is found either by Bayesian optimization or by an iterative method. These methods have been further improved by Nielsen [3] for better support of sailing ships and to improve inference speed [4]. These methods use known or estimated response functions of the ship to generate the expected spectrum.

Data driven methods, that do not depend on any knowledge of the ship, can be used when enough data has been measured [5]. This limits its applicability in two ways. First of all, such data is scarce. Not only are the ship motions needed, also a measured sea state is required. Secondly, this approach is only available for the ship the measured data came from. However, the approach

*Equal contribution

with neural networks from [5] allows generalization over multiple ships and multiple data sources. In this study, generalization over multiple ship geometries was investigated, using simulated data. The results presented show very good generalization.

The ideal solution would only use such a range of simulated data, removing the need for any measured data, other than for validating the method. This aspect is also investigated in this study. Although the proposed method does not work well enough, further investigation shows that the features that are used from the simulated data are compatible with the features present in measured data. From this, some promising future work is proposed.

2 MACHINE LEARNING

In this study, neural networks were used to infer sea state information from the ship motions. Simulated data is used to train the neural networks, making it a data driven solution, even though the data was generated using a physics model. The data that was generated is time domain ship motion data (6DOF), which captures local phase differences as well, as opposed to frequency domain data. This way, a dataset is built with ship motion time series and the corresponding sea state. Details about the dataset can be found in Section 3.1.

Some neural networks, especially Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are useful for time series regression. Specific neural network structures were designed for the problem at hand. Detailed discussion on motivation and designs can be found in [5]. The next few sections briefly describe the used networks.

2.1 CNN-REG

The convolutional neural network for regression (CNN-REG) was considered as a base network for comparison, see Fig. 1 for a diagram of the network. It comprises two convolution layers, each of which has the tanh activation function and is followed by a max pooling layer. After the output of the second max pooling layer is flattened, it is connected to a dense layer with the tanh activation followed by a dropout. The output of the dropout is sent to the output layer with tanh activation. Both convolution

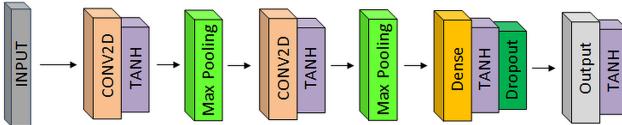


FIGURE 1: The architecture of the CNN-REG network.

layers have 48 filters. The filter sizes in the first two layers are

6×15 and 1×9 , respectively, and the pooling size is 3. The dense layer has 30 nodes, and the dropout rate is 0.25.

2.2 Multivariate LSTM-CNN (MLSTM-CNN)

MLSTM-CNN network was mainly adopted from [6]. It consists of a fully convolutional block and an LSTM block as shown in Fig. 2. The depth of the LSTM layer is 8. The first convo-

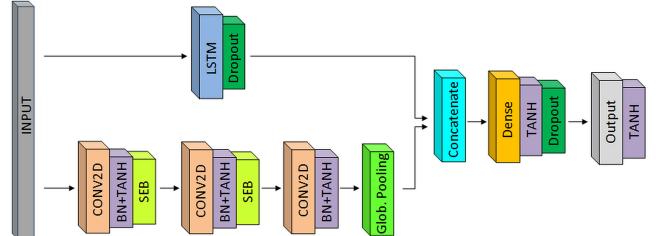


FIGURE 2: The architecture of the MLSTM-CNN network.

lution layer has 16 filters of size 6×11 . The second and third convolution layers have 32 filters and sizes of 1×6 and 1×3 , respectively. The dense layer has 8 nodes and the dropout rate is 0.1.

2.3 Sliding Puzzle Network

The Sliding Puzzle Network is designed to respond to individual features, without temporal relations between them. This is achieved by selecting patches at random locations from an input sample that have the same length as the filters in the network. Position independence is achieved by reducing filter activations to statistics, in this case mean, minimum and maximum. The diagram for this network is shown in Fig. 3.

The network uses 64 filters in the temporal direction, with size 1×25 , followed by 128 filters that combine the 6 channels, with size 6×1 . The dense layer has 30 nodes and no dropout is used.

3 DATA SOURCES AND TREATMENT

Two types of data were used, namely simulated data and measured data. The details of the two datasets are explained in the next three sections.

3.1 Setup of the numerical simulations

Numerical simulations were carried out in 6-DOF with a time-domain seakeeping and manoeuvring tool called FREDYN v16.1.1 [7, 8]. A strip theory based seakeeping tool in the frequency domain named SHIPMO v17.2.2 was used as a preprocessor to calculate added mass and damping coefficients and

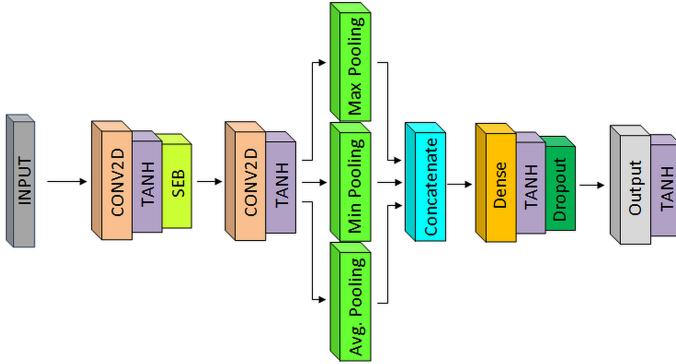


FIGURE 3: The Sliding Puzzle network. Note that two convolutional layers are used instead of one. The first convolutional layer only works in the time direction, the second combines the 6 channels. The receptive field of this combination is the same as the receptive field of a single convolution with combined dimensions. The number of trainable weights is less though, which leads to better generalization.

diffraction forces. The time step size was 0.25s, and the fourth-order Runge-Kutta time integration scheme was chosen. The ships in all the simulations were controlled in *heading* mode; the autopilot maintained the heading of the vessel rather than its track. The simulations were carried out at a ship speed of 15 knots for a duration of 1800 seconds in 6-DOF.

Hs [m]	Tp [s]															Sum
	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5		
< 1	0	47	97	154	140	103	58	23	9	3	1	0	0	0	0	635
1 – 1.5	0	45	121	244	275	227	143	61	27	9	3	0	0	0	0	1155
1.5 – 2	0	37	115	261	334	303	211	105	50	19	6	1	0	0	0	1442
2 – 2.5	0	23	78	205	318	329	264	153	78	33	10	3	0	0	0	1494
2.5 – 3	0	13	50	149	272	309	272	173	94	42	14	5	0	0	0	1393
3 – 3.5	0	7	30	93	195	242	235	165	96	46	18	7	0	0	0	1134
3.5 – 4	0	3	17	53	133	181	191	146	90	45	19	8	1	0	0	887
4 – 4.5	0	1	11	31	84	125	139	116	75	39	17	8	3	0	0	649
4.5 – 5	0	0	6	17	51	81	97	88	58	32	15	7	4	0	0	456
5 – 5.5	0	0	2	11	33	51	64	61	38	24	13	5	4	0	0	306
5.5 – 6	0	0	0	6	19	30	41	41	25	18	11	4	3	0	0	198
6 – 6.5	0	0	0	2	9	18	27	27	19	14	9	4	1	0	0	130
6.5 – 7	0	0	0	0	4	10	17	17	14	10	7	3	0	0	0	82
> 7	0	0	0	0	4	6	11	11	10	6	5	1	0	0	0	54
Sum	0	176	527	1226	1871	2015	1770	1187	683	340	148	56	16	0	0	10015

FIGURE 4: Global wave statistics - North Pacific scatter diagram.

Figure 4 shows a North pacific scatter diagram. From this list, any sea state (Hs, Tp) with the number of observations less than 5 was ignored. Also the sea states with Tp= 16.5s were ignored because of the low average number of observations. In the end, the list of sea states enclosed by the red line was considered as uniformly distributed and as such used in the simulations. For each sea state, 60 wave directions were considered by uniformly splitting the range of 0° – 354° at 6° interval. The combinations

of Hs, Tp and Dp resulted in a total of 7140 simulations per ship. The waves were realized using Jonswap spectrum with $\gamma = 3.3$, and the spectrum was discretized using 80 Fourier components. Table 1 lists the ranges of the main particulars for three groups of ships: *large*, *medium* and *small*. These groups were formed after ordering the ships according to their LBPs. In total, 11 different geometries were used with three of them having two loading conditions, and thus resulting in a total of 14 sets of simulations. The group called *large* included 3 different geometries two of them having two different loading conditions. The group called *medium* included 4 different geometries one of them having two loading conditions. The group named *small* consisted of four different geometries. 7140 simulations per ship resulted in a total of 99960 simulations for the 14 ships. Neural networks were trained on the three groups in various combinations. More details about this will be given in Section 4.

TABLE 1: Main particulars of the ships used in the numerical simulations.

Name	LBP [m]	Beam [m]	L/B [-]	Δ	GM [m]
Large	[124, 142]	[14, 19]	[7.4, 8.4]	[4000, 9000]	[1, 2.2]
Medium	[102, 120]	[12, 17]	[6, 9]	[2000, 4500]	[0.5, 1.6]
Small	[52, 75]	[9.5, 11.5]	[5.5, 6.5]	[300, 1634]	[0.48, 1.66]

3.2 Treatment of the numerical simulation results

The results of the numerical simulations were first investigated. Any simulation where capsizing occurred was ignored. The surge, sway and yaw motions were filtered to be divided in a wave-frequent (WF) and a low-frequency (LF) part.

The input to the neural networks was a multivariate time series $X \in \mathbb{R}^{6 \times d \times 1}$, where 6 refers to the 6-DOF motions and d is the length of the motion signals, from which samples were extracted at 0.75 second intervals. The value of d and the effective sample time were obtained as a result of a study where the effect of the duration and the sampling rate of the motion signals on the performance of the neural networks was examined. In the end, d was chosen to be 200, and the resulting signal duration was 2.5 minutes. For each ship 7140 samples were extracted, one sample per simulation.

3.3 Treatment of the in-service measurement data

The in-service measurement data was collected for a period of two years on board a frigate-type naval vessel. The 6-DOF mo-

tions at the center of gravity of the ship were measured using accelerometers, and the wave characteristics were measured via a wave scanning radar mounted on the ship. The data was saved as 30 minutes long pieces. The sampling rate of the ship motions was 20 Hz. The sampling rate of the relative wave direction and ship speed was 1 Hz. Hs and Tp were recorded as two-minutes averages. The in-service measurement data required preprocessing, selecting parts with steady behavior and resampling to 0.75 second intervals, and was unbalanced. Furthermore, the ground truth, i.e. wave characteristics, contained contributions from wind and swell seas and possibly other sources such as currents. Similar to the numerical simulation results discussed in Section 3.2, the input to the neural networks was a multivariate time series $X \in \mathbb{R}^{6 \times d \times 1}$ where d was chosen to be 200, and the resulting signal duration was 2.5 minutes. In total, 20120 samples were extracted from the in-service measurement data. Figures 5 and 6 illustrate the mean and standard deviation of each sample for 6-DOF ship motions, while Fig. 7 illustrates the Hs, Tp, Dp and Vs for each sample. Note that the samples were chronologically ordered in the sense that the sample with the index value of n was collected later than the sample with index $n - 1$, and earlier than the sample with index $n + 1$. Figure 8 shows the histogram of Dp, where the unbalanced nature of the data can be clearly observed.

4 RESULTS

4.1 Training methodology

We have trained our networks with stochastic gradient descent (SGD) [9] utilizing the Keras library [10] with the TensorFlow-GPU backend [11] running on a NVidia GeForce GTX 980 with 2048 CUDA cores and 4GB memory. The processor was the Intel Xeon CPU E5-1630 v4 with 3.7 GHz. In SGD, we used a momentum of 0.9 with a decay of 0. The loss function was the mean squared error (MSE). We used different learning rates with different decays for the three neural networks, which will be mentioned in the corresponding sections below.

For each neural network, a hyperparameter tuning study was carried out to optimize the performance and efficiency of the networks. Since the architecture of the networks were different, the hyperparameter tuning study varied between the networks in required effort.

4.2 Evaluation metrics

The results are reported as the 95% error level. This means that 95% of the predictions errors is less than or equal to this value.

4.3 Results from the numerical simulations

Table 2 lists the performance of the three neural networks, namely CNN-REG from Section 2.1, MLSTM-CNN from Section 2.2, and Sliding Puzzle from Section 2.3, for numerical simulation results. The values in the table indicates the average of

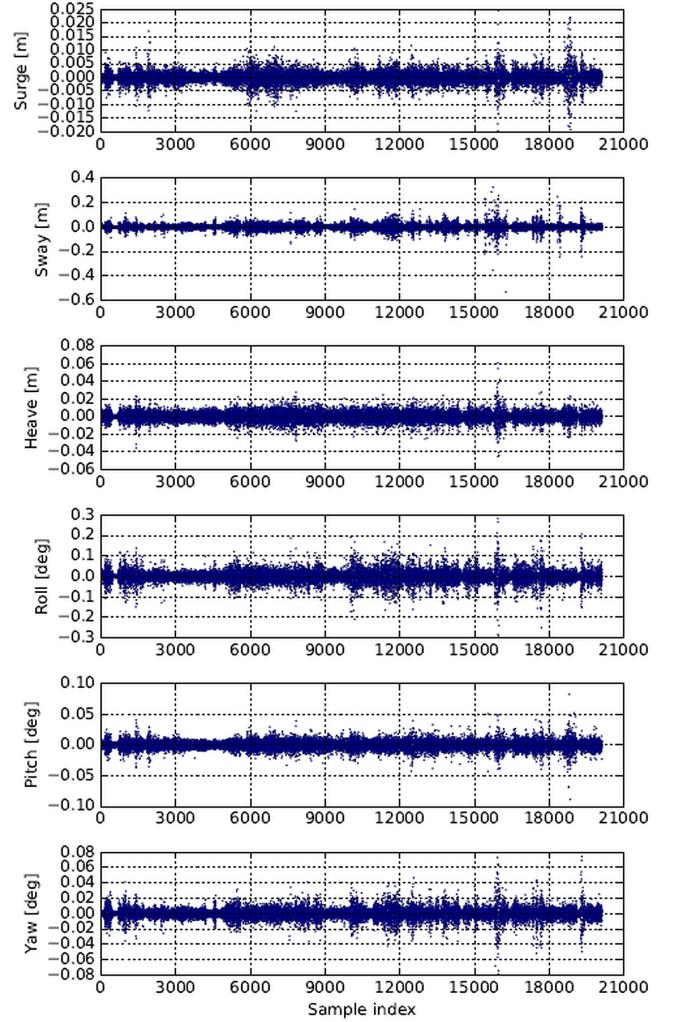


FIGURE 5: Mean of the ship motions of each sample from the in-service measurement data.

the 5-fold cross validation. The demonstrate that the 95% levels for validation for all the data sets were in the single digits; from MLSTM-CNN and Sliding Puzzle the values were around 1 deg, while from CNN-REG they were around 4 deg. This underlines the main strength of the adopted approach in estimating Dp from ship motions regardless of the ship geometry. In terms of test data performance, it can be observed that the 95% levels were again very good especially from the MLSTM-CNN and Sliding Puzzle networks. This highlights the capability of the adopted approach in generalizing over ship geometry. Regarding computational effort for training, it can be seen that the CNN-REG is the least expensive and MLSTM-CNN and is the most expensive network.

Figure 9 shows the convergence of the training and validation losses from the three neural networks which were trained and

TABLE 2: Results from the three neural networks using various combinations of the numerical simulation data. Note that the letters *L*, *M* and *S* stand for *Large*, *Medium* and *Small*, respectively, indicating the groups of ships from Tab. 1. Note that for some datasets, the 95% error levels for validation are not given. Since the neural networks were already trained and validated on those datasets, the trained models were reused for inference on the new test data. Therefore, only the statistics on the test error were listed for those datasets.

Train./Valid. data	Test data	95% level valid. [deg]	95% level test [deg]	Mean of test error [deg]	Std of test error [deg]	Network name
L	S+M	0.94	16.92	-0.33	11.87	MLSTM-CNN
		1.02	9.95	0.03	4.72	Sliding Puzzle
		4.24	35.83	0.45	16.84	CNN-REG
M	S+L	0.90	9.45	0.0	4.42	MLSTM-CNN
		1.05	9.71	0.02	4.55	Sliding Puzzle
		4.08	11.43	-0.02	5.60	CNN-REG
S	M+L	1.00	10.20	0.27	9.42	MLSTM-CNN
		1.40	10.53	-0.36	5.43	Sliding Puzzle
		4.09	12.56	-0.19	9.18	CNN-REG
L	M	-	11.42	0.34	8.0	MLSTM-CNN
		-	5.94	0.04	2.76	Sliding Puzzle
		-	32.44	-0.14	15.35	CNN-REG
L	S	-	21.19	-1.14	15.40	MLSTM-CNN
		-	13.35	0.02	6.42	Sliding Puzzle
		-	38.36	1.23	18.55	CNN-REG
M	L	-	6.92	0.01	3.30	MLSTM-CNN
		-	6.18	0.04	2.95	Sliding Puzzle
		-	9.22	-0.12	4.48	CNN-REG
M	S	-	10.72	-0.03	5.50	MLSTM-CNN
		-	12.10	0.0	6.01	Sliding Puzzle
		-	13.80	0.09	6.77	CNN-REG
S	L	-	10.39	0.31	9.97	MLSTM-CNN
		-	12.03	-0.40	6.23	Sliding Puzzle
		-	11.67	0.0	10.10	CNN-REG
S	M	-	9.93	0.23	8.75	MLSTM-CNN
		-	8.90	-0.30	4.51	Sliding Puzzle
		-	13.37	-0.38	8.15	CNN-REG
L+M	S	1.17	9.84	0.01	5.27	MLSTM-CNN
		1.07	9.11	0.03	4.49	Sliding Puzzle
		4.14	14.12	-0.34	6.70	CNN-REG
L+S	M	1.36	5.97	0.16	2.80	MLSTM-CNN
		1.25	4.40	-0.02	2.14	Sliding Puzzle
		4.58	9.66	-0.01	4.71	CNN-REG
M+S	L	1.43	5.95	-0.16	3.18	MLSTM-CNN
		1.31	5.62	0.09	2.89	Sliding Puzzle
		4.42	8.90	0.0	4.37	CNN-REG

validated on the combined long and medium ships (L+M), and tested on the short ships (S), see the tenth row in Tab. 2. It can be seen that the MLSTM-CNN and Sliding Puzzle networks perform similarly and much better than the CNN-REG network. Training and validation losses are similar from the three networks indicating that overfitting was not observed.

Scatter plots of the truth and prediction values from the three net-

works are shown in Fig. 10 on the combined long and medium ships (L+M) data. The three networks exhibit an almost uniform performance in terms of prediction accuracy over the entire range of Dp values between -180 deg. to 180 deg. Between the three, it can be seen that the MLSTM-CNN and Sliding Puzzle networks have smaller scatter around the truth values indicating better prediction accuracy compared to the CNN-REG network.

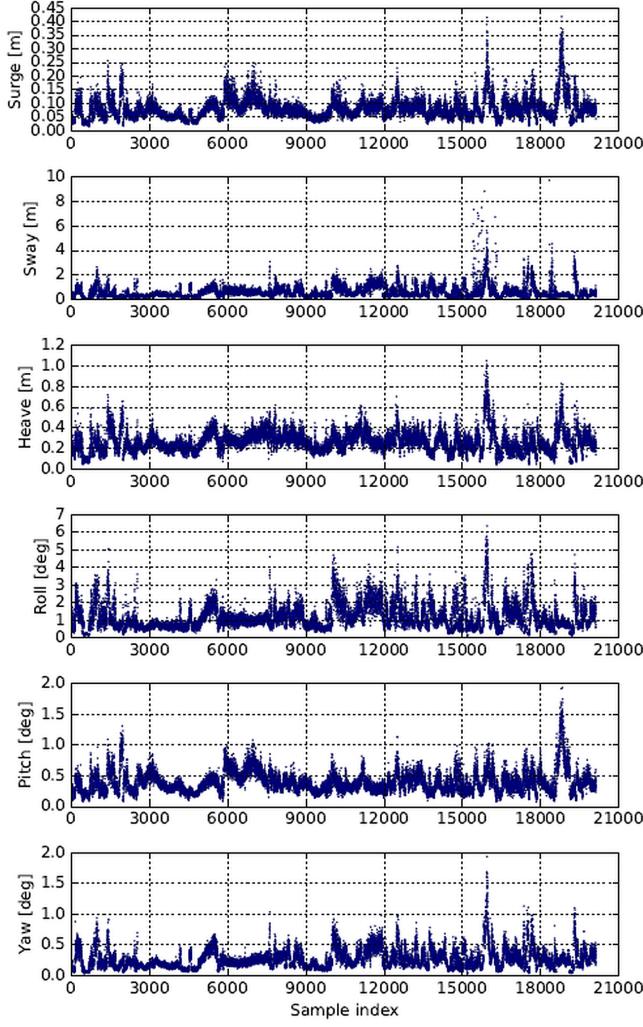


FIGURE 6: Standard deviation of the ship motions of each sample from the in-service measurement data.

TABLE 3: Results from the three neural networks applied with the direct application and transfer learning approaches.

Train./Valid. data	95% level	Mean of error	Std of error	Network name
	[deg]	[deg]	[deg]	
Direct application	82.52	-8.36	43.97	MLSTM-CNN
	140.42	-4.40	61.33	CNN-REG
	68.44	-2.04	35.42	Sliding Puzzle
Transfer learning	33.22	-0.13	16.83	MLSTM-CNN
	45.64	0.11	25.61	CNN-REG
	31.77	0.17	19.05	Sliding Puzzle

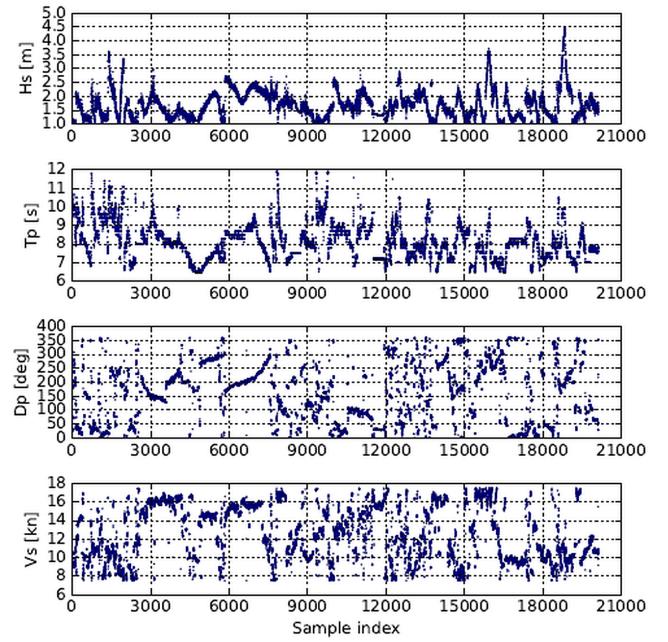


FIGURE 7: Hs, Tp, Dp and Vs of each sample from the in-service measurement data.

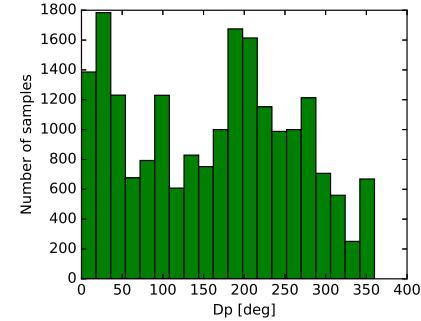


FIGURE 8: Histogram of Dp from the in-service measurement data.

Figure 11 illustrates the histograms of the training and validation errors from the three networks on the same data. The histograms from the CNN-REG network are wider and flatter, while the ones from the MLSTM-CNN and Sliding Puzzle have narrow sharp peaks around the value of zero.

Figure 12 demonstrates the histograms of the test error and scatter plots of the truth and prediction on the test data from the three networks trained on the combined long and medium ships (L+M) data. It can be observed that the prediction accuracy of the networks is not uniform over the entire range of the Dp values. The filters learned on the long and medium ships are more useful for some Dp values on the short ships. This is expected as the low

frequency response learned for the short ships is still useful for the longer ships, but the higher frequencies can only be learned from the dynamics of the shorter ships. Especially the values from the MLSTM-CNN and Sliding Puzzle networks show the generalization capability of the neural networks.

The results in Tab. 2 show that the neural networks can generalize quite accurately over the geometry. Between the three networks, the MLSTM-CNN and Sliding Puzzle in general perform similarly and better than the CNN-REG, but Sliding Puzzle exhibits a superior performance in terms of accuracy. The difference between the MLSTM-CNN and Sliding Puzzle becomes more pronounced when they are trained and validated on the long ships (L) and tested on other groups and combinations of ships. In those cases, 95% level for the prediction on the test data is much larger with the MLSTM-CNN than with Sliding Puzzle.

4.4 Reusing the models trained on simulated data for the measured data

In this section, results are presented from the three networks, namely CNN-REG, MLSTM-CNN and Sliding Puzzle, that were trained on the medium ships (M) and reused on the in-service measurement data (note that detailed results from the neural networks trained on the measured data are given in [5]). These networks were selected as they are the worst- and best-performing networks considering the results from the previous section. Two approaches were adopted when reusing the models on the measured data:

Transfer learning: Here, the decoder layers (the last two dense layers in the networks) of the trained models were further trained on the in-service measurement data. Note that apart from the decoders, the learned parameters of all the other layers were not changed. Transfer learning allows using the learned features of the model trained on the simulated data, and repurposing these features to a second target model to be trained on the in-service measurement data. If the learned features are general in the sense that they are suitable on both data sets, then this approach is expected to work properly.

Direct application: Here, the models trained on the simulated data were directly applied on the in-service measurement data without changing any parameter of the model.

Figure 13 illustrates the results from the three networks when the direct application approach is adopted, while Figure 14 illustrates the results with the transfer learning. Table 3 lists the 95% level from the three networks with the two approaches. It can be observed that the transfer learning approach results in considerably better results compared to the direct application approach. This indicates that the features learned by the filters on the simulated data can be indeed used with the measured data if they are

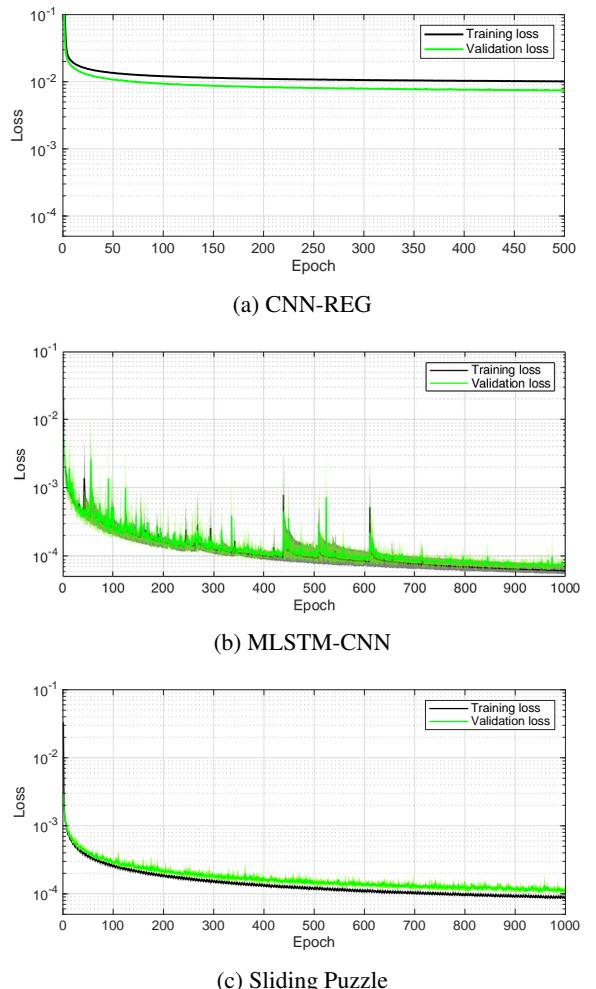


FIGURE 9: Training and validation losses from the three networks on the combined long and medium ships (L+M) data, see the tenth row in Tab. 2.

decoded correctly. Between the neural networks, Sliding Puzzle shows a superior performance with both approaches. The validation error of 31.77 deg from the Sliding Puzzle network is an acceptable result opening the door for further studies with numerical simulation, even though it is not as good as the results reported in [5]. Clearly there is also a difference in the features captured by the trained filters.

5 CONCLUSIONS

In this study, machine learning approaches were employed to estimate relative wave direction from time histories of ship motions. For that purpose, an extensive numerical simulation campaign was carried out in 6-DOF with a time-domain seakeeping and manoeuvring tool. Eleven different ship geometries

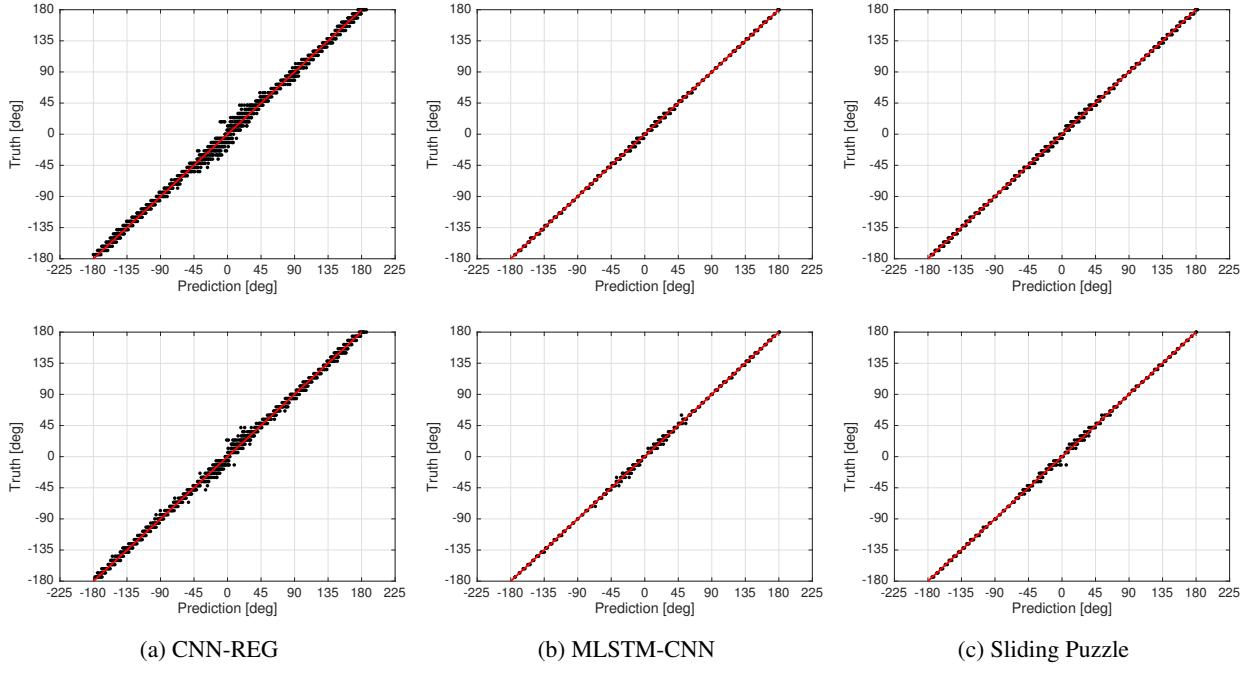


FIGURE 10: Truth and prediction values from the three networks on the combined long and medium ships (L+M) data, see the tenth row in Tab. 2. The plots on the top row illustrate the results for the training data, and those on the bottom row validation data.

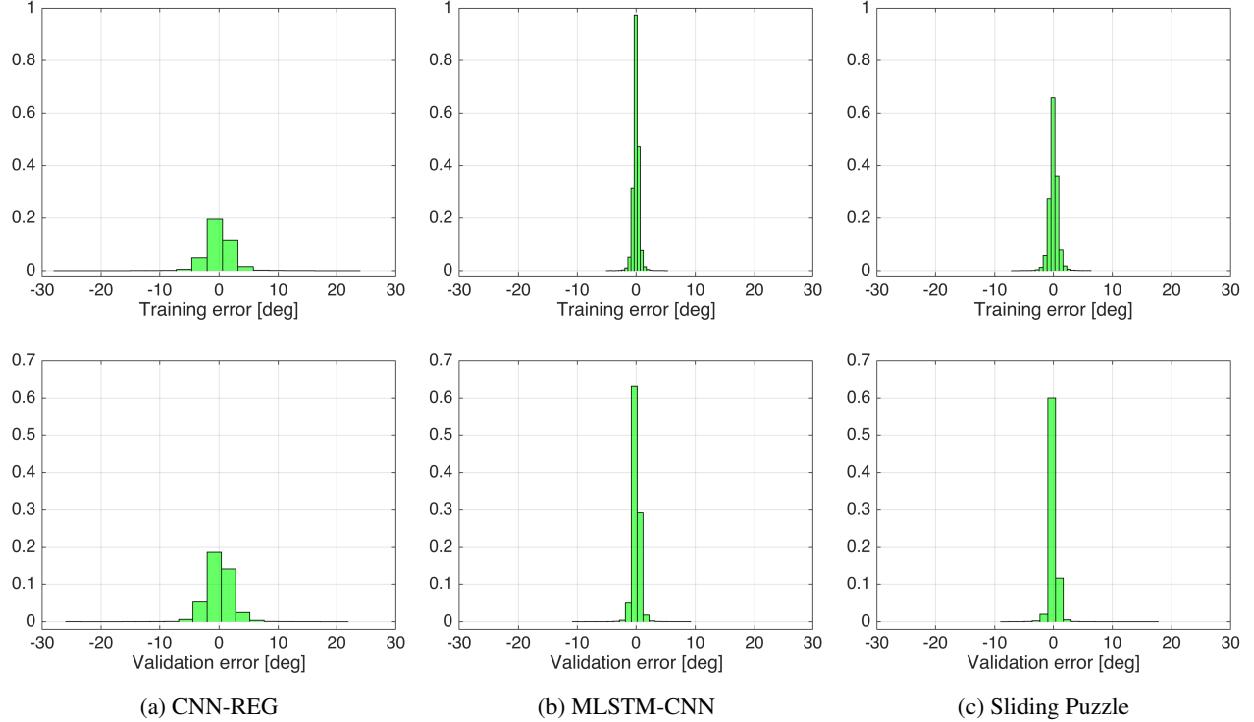


FIGURE 11: Histograms of the training and validation error from the three networks on the combined long and medium ships (L+M) data, see the tenth row in Tab. 2.

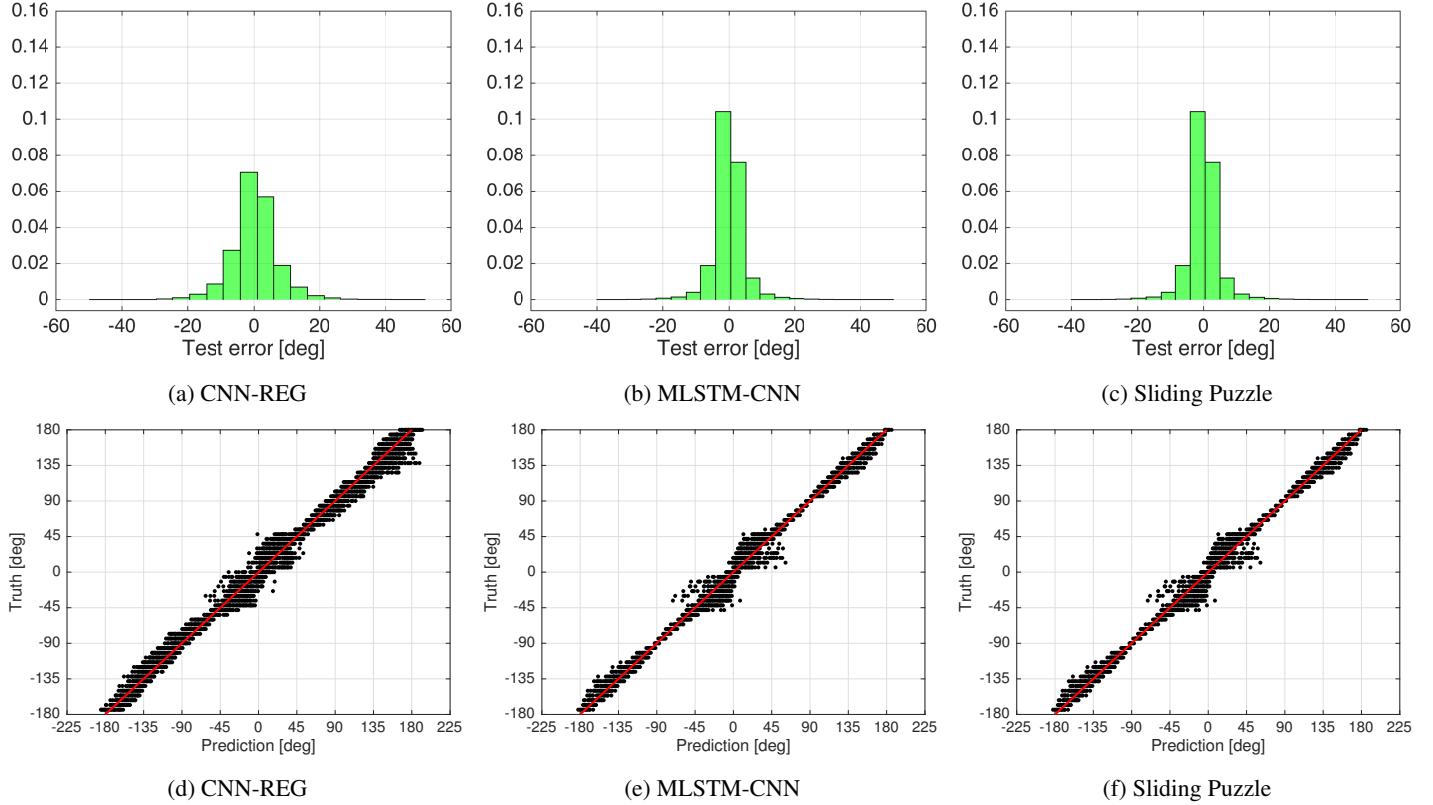


FIGURE 12: Histograms and scatter plots for the test data of the short ships (S) from the three neural networks trained on the combined long and medium ships (L+M) data, see the tenth row in Tab. 2.

were used in the simulations, and 7140 simulations were performed per ship taking a comprehensive set of combinations of H_s , T_p and D_p . Various neural network architectures were considered and their performances were compared. Furthermore, the networks trained on the simulated data were reused for the in-service measurement data by adopting two approaches, *direct application* and *transfer learning*. The performance of the neural networks with these approaches were also investigated.

The results indicate that the adopted approach in this study shows a very good performance in estimating relative wave direction from time histories of 6-DOF ship motions. By including convolutional filters to encode the phase relations between the 6-DOF motions, the adopted neural networks were capable to estimate the D_p accurately. Furthermore, they were able to generalize over the geometry very successfully opening the door for interesting future topics.

Finally, the neural networks trained on the simulated data were reused on the in-service measurement data collected over the period of two years on board a frigate type vessel. The results show that the learned features of the model trained on the simulated data were indeed suitable for the measured data as well. Especially when the transfer learning approach was adopted, the

neural networks were able to estimate D_p accurately. Future work will include more complex numerical simulation studies with short-crested waves and combined wind and swell sea spectrums. The neural networks will also be trained to estimate the full directional spectrum of the waves (encoder-decoder networks).

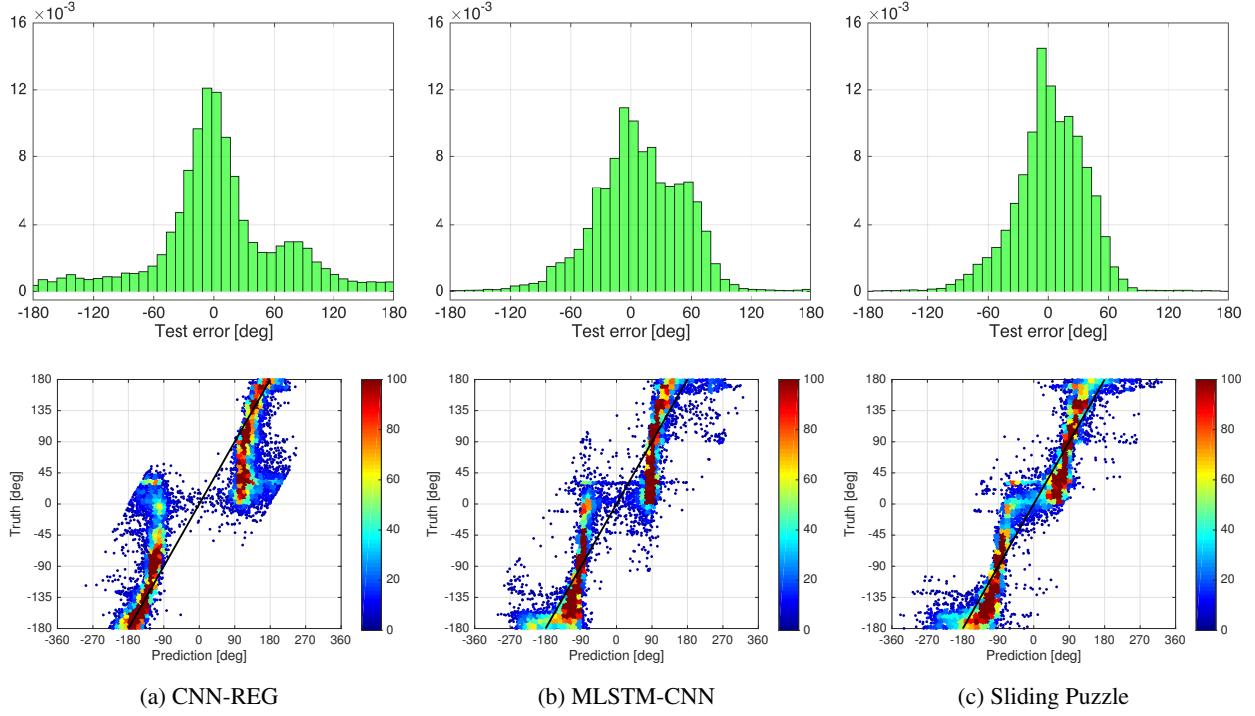


FIGURE 13: Histograms and scatter plots from the three networks with the direct application approach.

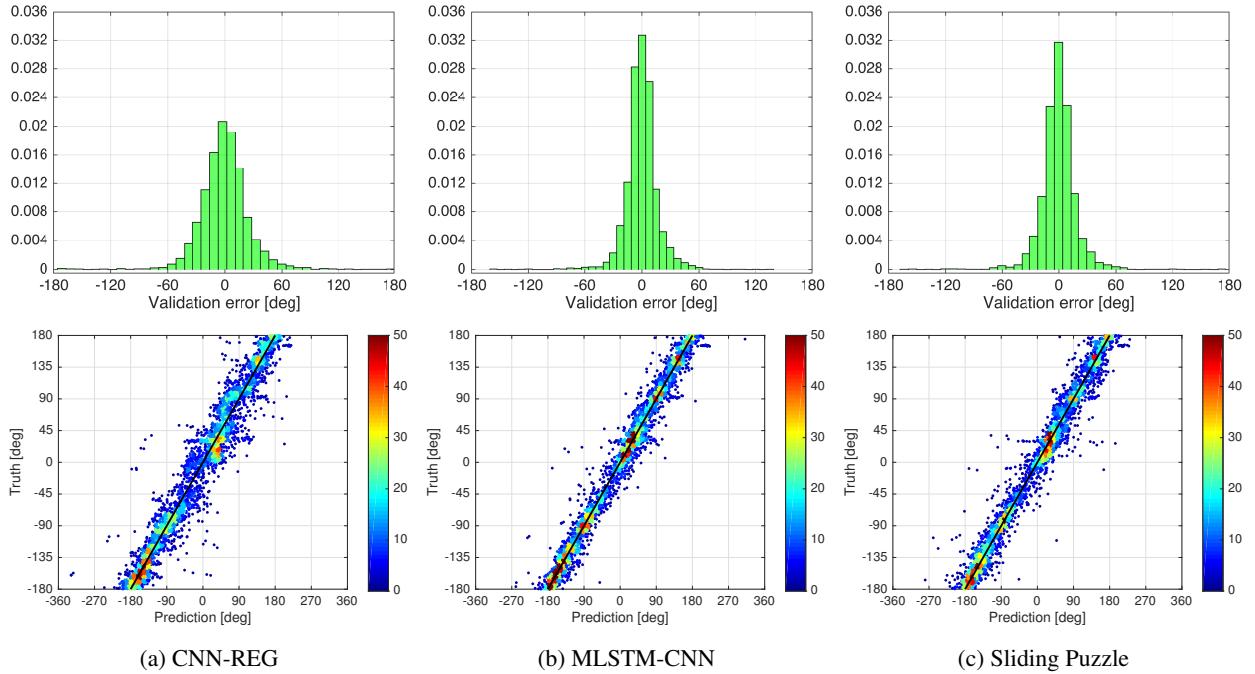


FIGURE 14: Histograms and scatter plots from the three networks with the transfer learning approach.

REFERENCES

- [1] Iseki, T., and Ohtsu, K., 2000. “Bayesian estimation of directional wave spectra based on ship motions”. *Control Engineering Practice*, **8**, 02, pp. 215–219.
- [2] Tannuri, E. A., Sparano, J. V., Simos, A. N., and Cruz, J. D., 2003. “Estimating directional wave spectrum based on stationary ship motion measurements”. *Applied Ocean Research*, **25**, pp. 243–261.
- [3] Nielsen, U., 2006. “Estimations of on-site directional wave spectra from measured ship responses”. *Marine Structures*, **19**, 01, pp. 33–69.
- [4] Nielsen, U. D., Brodtkorb, A. H., and Srensen, A. J., 2018. “A brute-force spectral approach for wave estimation using measured vessel motions”. pp. 101–121. Exported from <https://app.dimensions.ai> on 2019/01/07.
- [5] Mak, B., and Düz, B., 2019. “Ship As A Wave Buoy - Estimating relative wave direction from in-service ship motion measurements using machine learning”. In Proc. ASME 2019 38th International Conference on Ocean, Offshore and Arctic Engineering OMAE2019, paper OMAE2019-96201.
- [6] Karim, F., Majumdar, S., Darabi, H., and Harford, S. “Multivariate LSTM-FCNs for time series classification”.
- [7] McTaggart, K., and De Kat, J. O., 2000. “Capsize risk of intact frigates in irregular seas”. *Transactions SNAME*, **108**, pp. 147–177.
- [8] Van’t Veer, R., and De Kat, J. O., 2000. “Experimental and numerical investigation on progressive flooding and sloshing in complex compartment geometries”. In Proc. of the 7th Int. Conf. on Stab. for Ships and Ocean Vehicles (STAB 2000), February, Launceston, Australia, pp. Vol. A, 305–321.
- [9] Kiefer, J., and Wolfowitz, J., 1952. “Stochastic estimation of the maximum of a regression function”. *Ann. Math. Statist.*, **23**(3), 09, pp. 462–466.
- [10] Chollet, F., et al., 2015. Keras.
- [11] Abadi, M., et al., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.