



Real Time Estimation of Local Wave Characteristics from Ship Motions Using Artificial Neural Networks

Bulent Duz^(✉), Bart Mak, Remco Hageman, and Nicola Grasso

Maritime Research Institute (MARIN), Wageningen, The Netherlands
{b.duz,b.mak,r.hageman,n.grasso}@marin.nl

Abstract. Wave characteristics such as significant wave height, peak period and main direction have an important effect on all aspects of ship operations. Yet, estimation of these characteristics in real time in an accurate, robust and efficient manner poses a significant challenge. Current approaches include using wave buoys, wave radars, weather forecasts and regression based on ship motion statistics. External sources are not always available or reliable, and wave radars are expensive. Statistics based methods work reasonably well for wave height, but fail to estimate peak period and relative wave direction.

In this study the attention is focused on estimating the wave characteristics from ship motions using artificial neural networks. The inputs to the neural networks were the time histories of 6-DOF ship motions, and the outputs were the wave characteristics. Using time series instead of statistics preserves phase differences between signals. Especially when estimating the relative wave direction, these phase differences encode essential features. Therefore, the problem was treated as multivariate time series regression.

Two different data sources were used with neural networks: in-service measurement data and numerical simulation data. In-service measurement data was collected from a frigate type vessel for a period of two years. The numerical simulation data was produced by using a time-domain seakeeping and manoeuvring software.

In this paper, we present methods to successfully implement neural networks for real time estimation of local wave characteristics. The paper will include details about the measurement data, numerical simulations, the networks architectures and their performances on the measurement and simulation data.

Keywords: Wave characteristics · 6-DOF ship motions · Multivariate time series regression · Machine learning

1 Introduction

Sea state information is important for operational decisions or advice on all activities at sea. Personal and environmental safety require correct interpretations of the actual conditions. Efficiency suffers from incorrect interpretations

as well. In the longer run, having statistics on encountered conditions helps in designing structures and ships. This results in higher levels of safety and more realistic lifetime expectations. Hence, many methods have been proposed in the past few decades to infer sea state information from indirect measurements.

Traditionally, sea state characteristics are measured using wave buoys or satellite data. These methods result in a rough estimation, limited by the distance from the buoys and the resolution of the satellite data. Occasionally the sea state is monitored using a wave radar. Although this gives reliable results, it is not used very often because of the high costs.

An alternative is to use the motions of the vessel itself to infer sea state information. Methods have been developed by Tannuri [1] and Iseki [2] matching the measured motion spectrum with a predicted spectrum. The matching spectrum is found either by Bayesian optimization or an iterative method. Both approaches use estimated response functions as mapping from sea state to ship motions. This idea has been further optimized by Nielsen [3] to include sailing vessels and more efficient sea state estimation [4].

In this study 6-DOF ship motion data is used to build a parameterized model for direct sea state estimation. No knowledge of the ship is used to build the model. Instead, a parameterized machine learning model is trained using time series data of the ship motions together with the known sea state in a supervised learning setup. Being able to capture time series data in this way allows the use of local phase differences in the motion signals for the sea state estimation. In this study it is shown that specific neural network structures can be used to do so and that the resulting models perform well in a wide variety of sea states.

Such a data driven method can be used when sufficient data of good quality is available for training. For numerical simulation data, this might not be an issue, but for measurement data it can be a prohibitive factor. This issue limits the applicability of a data driven method in two ways. First of all, such data is scarce. Not only are the ship motions needed, but also measured sea states are required. Secondly, since the model is trained for the ship the measured data came from, its performance on another ship might deteriorate. However, generalization over multiple ship geometries was investigated in [5] using simulated data, and very good results were obtained.

The ideal solution would only use 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 for this, further investigation shows that the features that are used from the simulated data are compatible with the features present in the 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 6-DOF ship motions. Both simulated and measured data were used to train the neural networks. The data was recorded in the time domain, which captures

local phase differences, as opposed to frequency domain data. This way, datasets were built with ship motion time series and the corresponding sea state. Details about the datasets can be found in Sect. 3.

Some neural networks, especially Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), 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 [6]. The next few sections briefly describe two networks used in this study.

2.1 Multivariate LSTM-CNN (MLSTM-CNN)

MLSTM-CNN network was mainly adopted from [7]. It consists of a fully convolutional block and a Long Short-Term Memory (LSTM) block as shown in Fig. 1. The depth of the LSTM layer is 16 with a dropout rate of 0.25. The first convolution layer has 128 filters of size 6×12 . The second and third convolution layers have 192 and 128 filters of size 1×1 , respectively. The dense layer has 128 nodes and the dropout rate is 0.25.

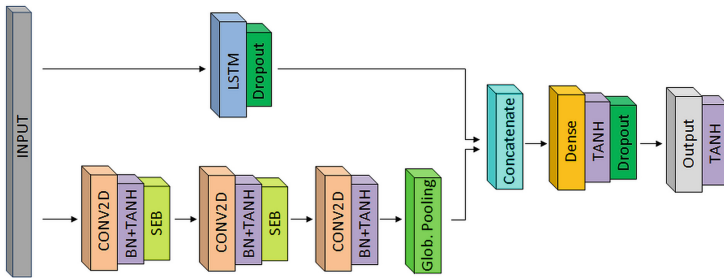


Fig. 1. The architecture of the MLSTM-CNN network.

2.2 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 of this network is shown in Fig. 2.

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.

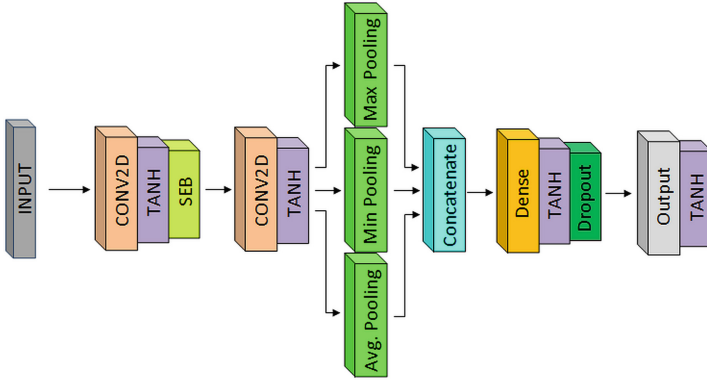


Fig. 2. The Sliding Puzzle network. Note that two convolutional layers are used instead of one. The first convolutional layer only works in the time direction, the second combines the 6 channels. The receptive field of this combination is the same as the receptive field of a single convolution with combined dimensions. The number of trainable weights is less though, which leads to better generalization.

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 [8,9]. 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 diffraction forces. The time step size was 0.25 s, and the fourth-order Runge-Kutta time integration scheme was used. The ship was 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 kn for a duration of 600 s. A frigate-type naval vessel was used as the geometry. The vessel was the same as the one on board of which the in-service measurement data was collected.

Ten thousand simulations were carried out for a wide range of significant wave height (H_s), peak period (T_p) and main direction (D_p). H_s , T_p and D_p were uniformly distributed over the ranges of [0.5 m, 10.5 m], [6.5 s, 15.5 s] and [0°, 360°), respectively. Latin hypercube sampling (LHS) was then used to sample 10000 points from this 3D input domain. Figure 3 shows the result from LHS. The waves were realized using Jonswap spectrum with $\gamma = 3.3$, and the spectrum was discretized using 80 Fourier components.

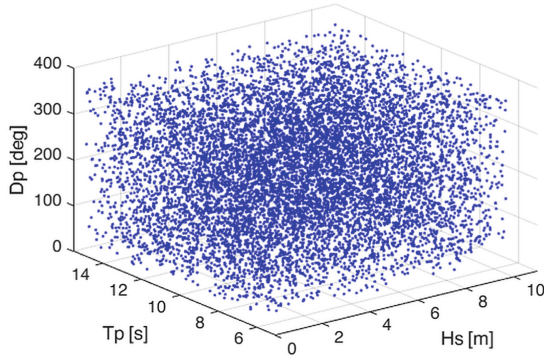


Fig. 3. Latin hypercube sampling (LHS) with 10000 points for the uniformly distributed Hs, Tp and Dp.

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-frequent (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. The value of d was 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 min. One sample was extracted from each simulation resulting in a total sample size of 10000.

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 motions 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 min long pieces. The sampling rate of the ship motions was 20 Hz, and the sampling rate of the ship speed was 1 Hz. The wave scanning radar provided two-minute data files which were used to derive statistical parameters, such as Hs, Tp and Dp. The accuracy of peak period and directionality of the wave was generally good. However, because the radar was mounted on a moving platform the wave height was not very accurate. In order to compensate this inaccuracy, a wave data fusion approach was executed to improve the wave height assessment [10]. The in-service measurement data required preprocessing and was unbalanced; Fig. 4 shows the histograms of Hs, Tp and Dp, where the unbalanced nature of the data can be clearly observed. More information about how the measured data was prepared for training is provided in [6]. Furthermore, the ground truth,

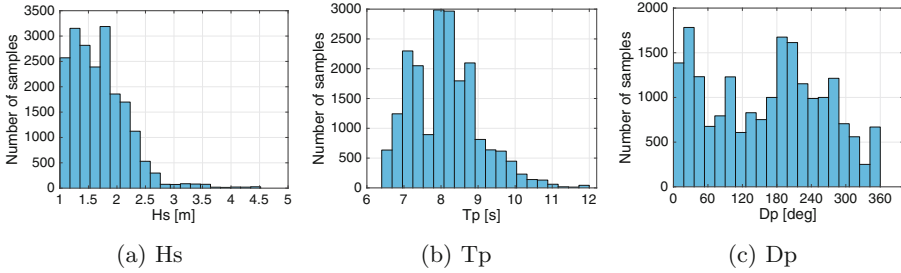


Fig. 4. Histogram of the wave characteristics from the in-service measurement data.

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 Sect. 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 min. 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 ship speed (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 $n + 1$.

4 Results

4.1 Training Methodology

We have trained our networks with stochastic gradient descent (SGD) [11] utilizing the Keras library [12] with the TensorFlow-GPU backend [13] and PlaidML [14] running on a NVidia GeForce GTX 1080 with 2560 CUDA cores and 8 GB 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). Learning rates of 0.06 and 0.03 were used with the simulated and measured data, respectively.

For each neural network, a hyperparameter tuning study was carried out to optimize the performance and efficiency of the networks. The hyperparameters included both network architecture parameters and training parameters, such as learning rates. Since the architecture of the networks were different, the hyperparameter tuning study varied between the networks in required effort.

The results are reported as the average (or combined in some cases) of the 5-fold cross-validation. In a cross-validation study, the original data is partitioned into a training set to train the model, and a validation set to evaluate it. In 5-fold cross-validation, the original data is partitioned into 5 equal size pieces. From the 5 pieces, a single piece is retained as the validation data to evaluate the

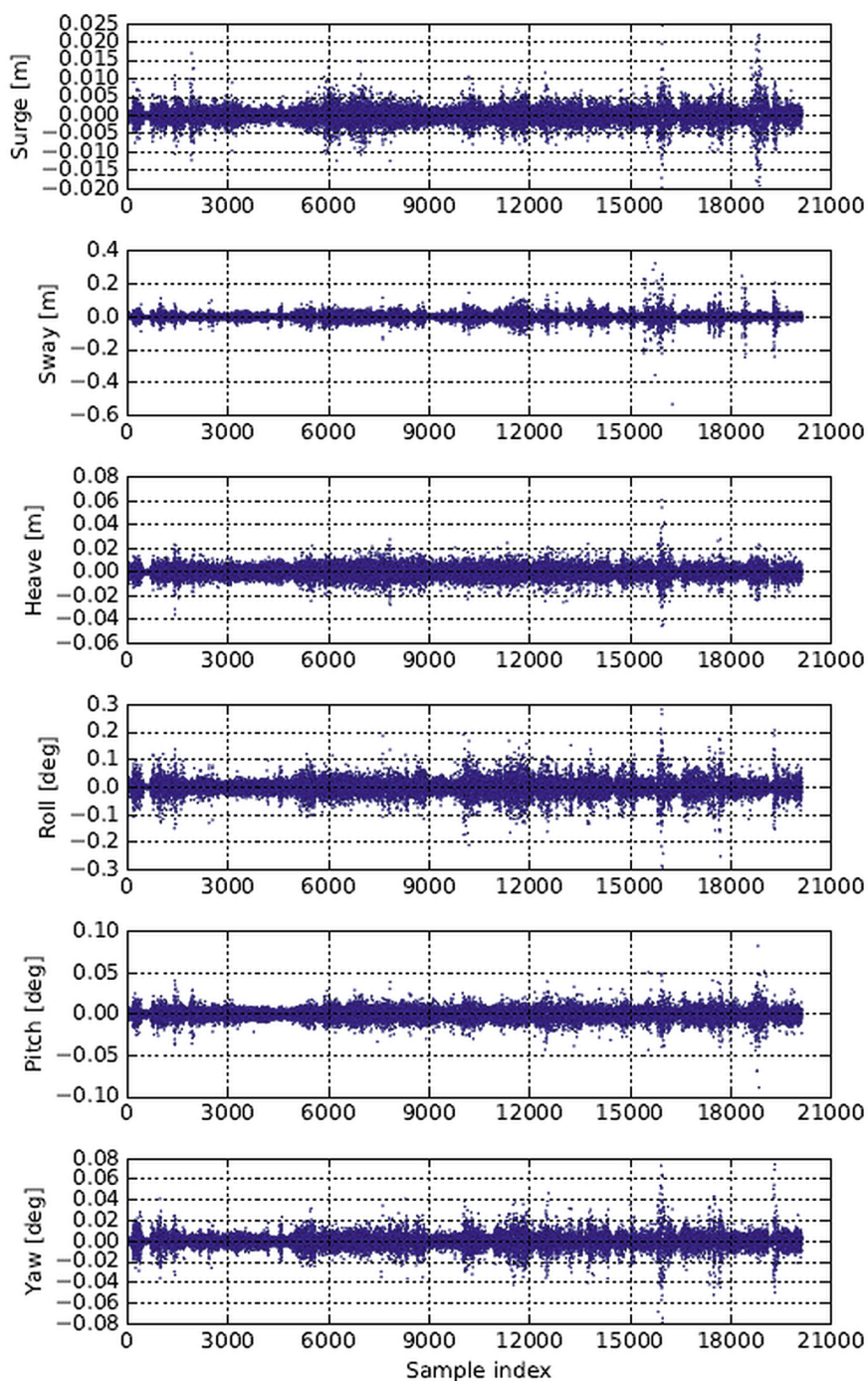


Fig. 5. Mean of the ship motions of each sample from the in-service measurement data.

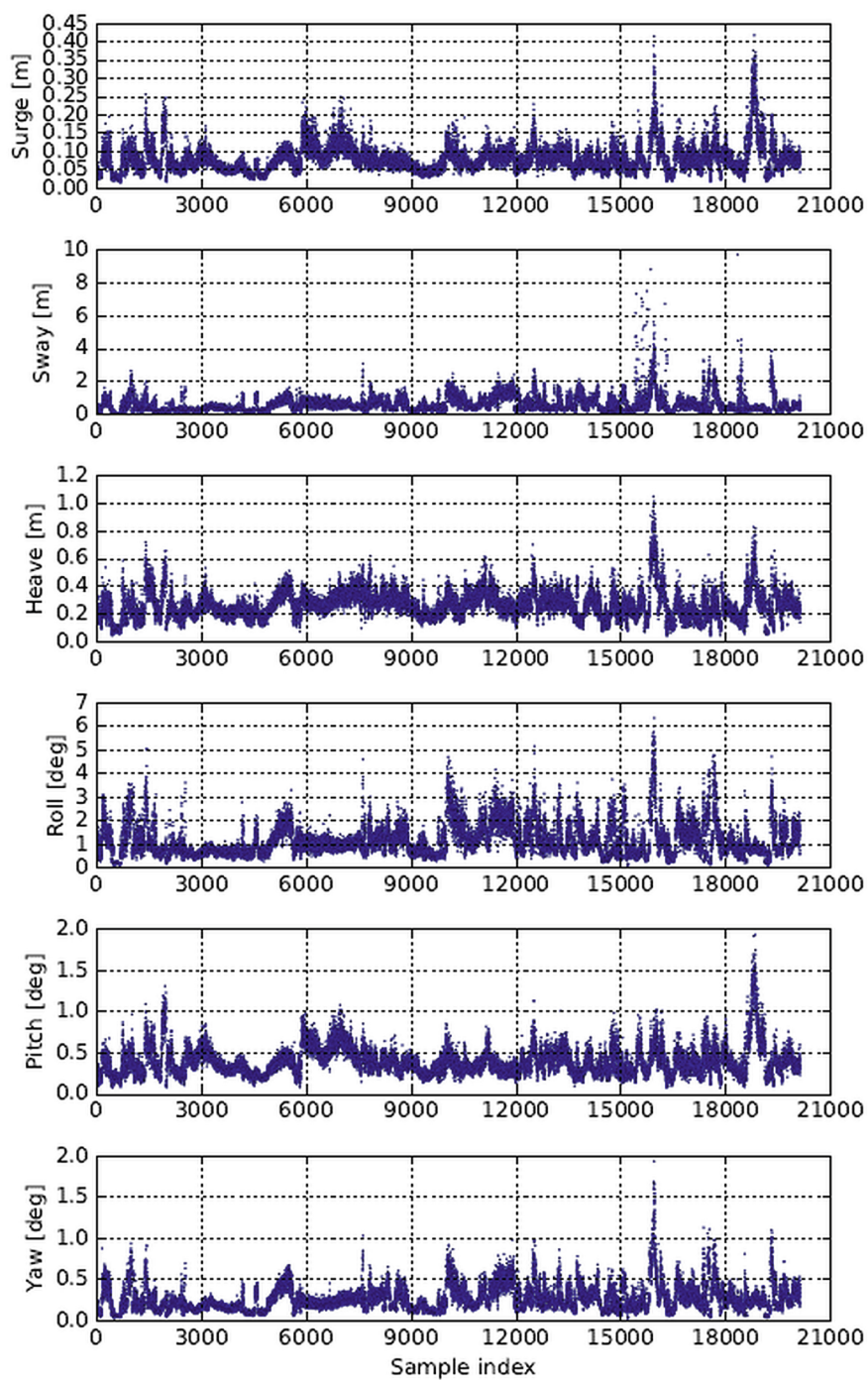


Fig. 6. Standard deviation of the ship motions of each sample from the in-service measurement data.

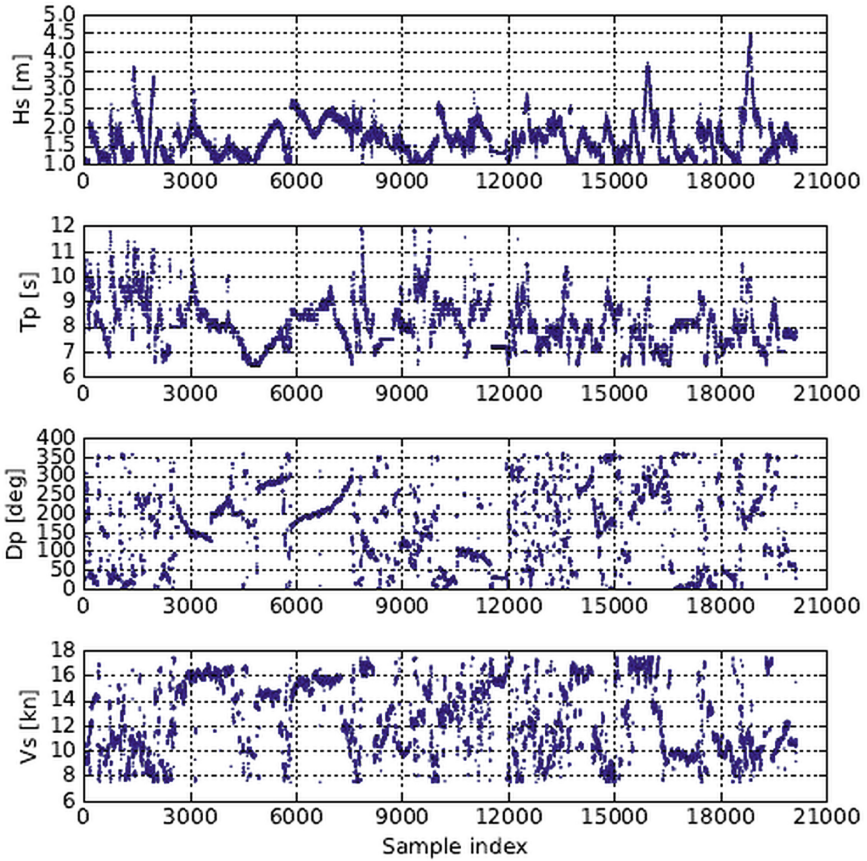


Fig. 7. Hs, Tp, Dp and Vs of each sample from the in-service measurement data.

model, and the remaining 4 pieces are used as training data. The cross-validation process is then repeated 5 times (the folds), which allows each of the 5 pieces being used exactly once as the validation data. The 5 results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all samples in the original data are used for both training and validation, and each sample is used for validation exactly once.

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. Additionally, mean and standard deviation of the prediction error are provided.

4.3 Results from the In-service Measurement Data

The 5-fold cross-validation with the measurement data was carried out in two fashions:

- Chronological: The original data was first ordered chronologically. Without shuffling, the 5-fold cross-validation procedure was then followed.
- Shuffled: The original data was shuffled first, and afterwards the 5-fold cross validation was carried out.

Figure 8 shows the convergence of the training and validation losses from the two neural networks for Dp. These losses are presented as the average and spread of 5-fold cross validation. The histograms in Fig. 9 show how the validation errors for Dp are distributed. It can be seen that the errors are distributed symmetrically, and the neural networks show better performance for the shuffled data. Figures 10 and 11 show the scatter plots of truth and prediction values on the shuffled and chronological measured data, respectively. Figure 12 illustrates the validation error of Dp with respect to Hs and Tp on the shuffled measured data using MLSTM-CNN. It can be seen that the validation error of Dp does not have any bias with respect to Hs or Tp. Similar observations were made for the validation errors of Hs and Tp as well. Table 1 lists the results for the evaluation metrics from the two networks applied on the shuffled and chronological measured data.

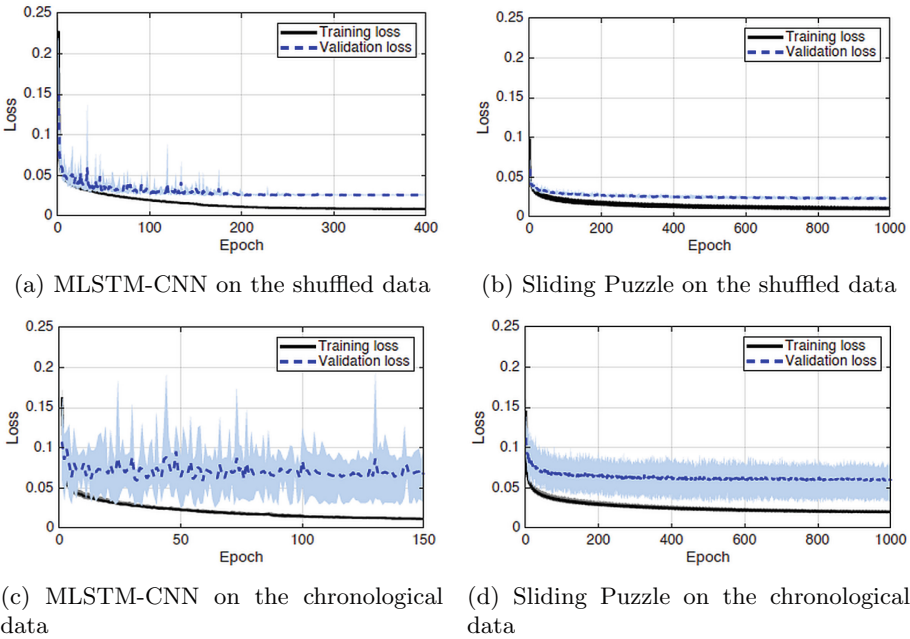
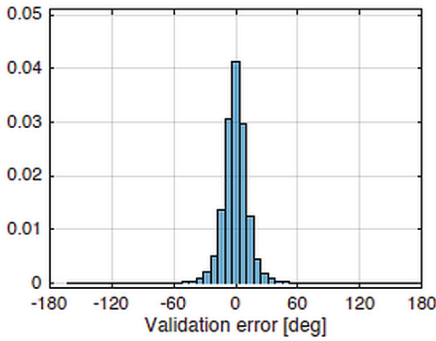


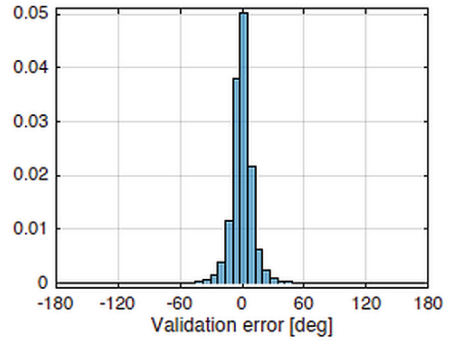
Fig. 8. Training and validation losses from the two networks on the shuffled and chronological measured data for Dp. Note the much larger spread in validation losses for the chronological data.

Table 1. Results from the two neural networks applied to the shuffled and chronological measured data.

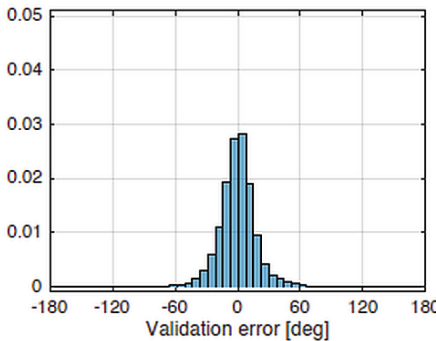
Train./Valid. data	Wave characteristic	95% level val.	Mean of val. error	Std of val. error	Network name
Shuffled	Dp [deg]	24.86	-0.05	13.59	MLSTM-CNN
		21.67	-0.20	11.71	Sliding puzzle
	Hs [m]	0.41	0.0	0.20	MLSTM-CNN
		0.39	0.0	0.19	Sliding puzzle
	Tp [s]	1.01	0.0	0.50	MLSTM-CNN
		0.95	0.0	0.46	Sliding puzzle
Chronological	Dp [deg]	37.46	-0.68	19.99	MLSTM-CNN
		38.29	0.53	19.35	Sliding puzzle
	Hs [m]	0.51	0.0	0.27	MLSTM-CNN
		0.51	0.0	0.27	Sliding puzzle
	Tp [s]	1.40	0.01	0.68	MLSTM-CNN
		1.48	0.01	0.69	Sliding puzzle



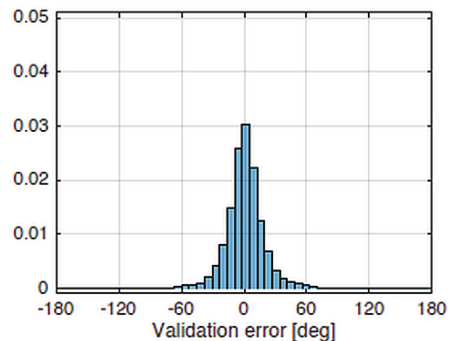
(a) MLSTM-CNN on the shuffled data



(b) Sliding Puzzle on the shuffled data



(c) MLSTM-CNN on the chronological data



(d) Sliding Puzzle on the chronological data

Fig. 9. Histograms of the validation errors from the two networks on the shuffled and chronological measured data for Dp.

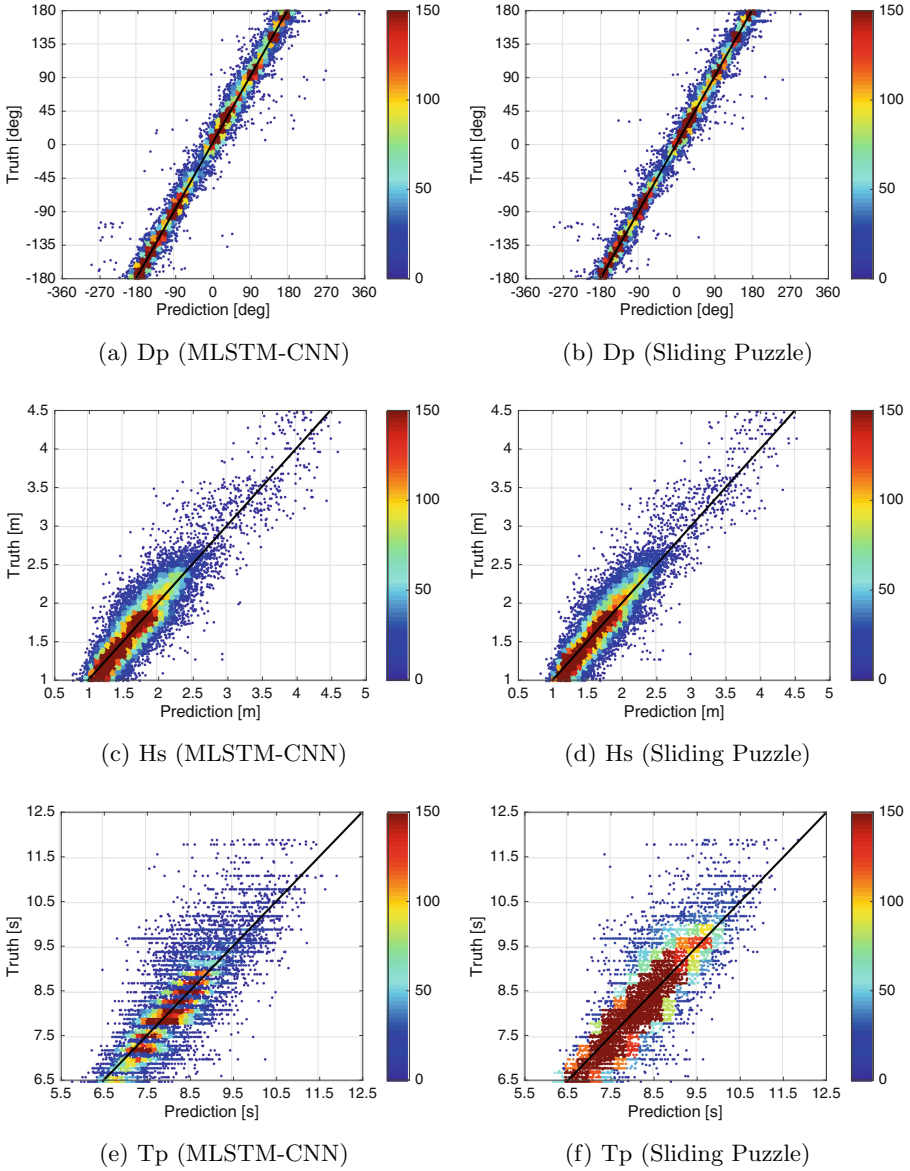
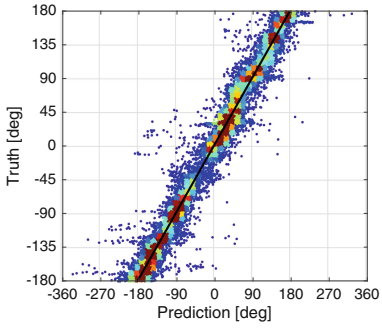
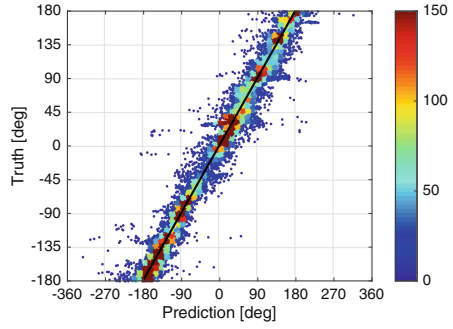


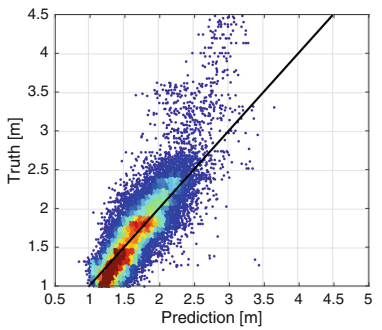
Fig. 10. Truth and prediction values for Dp, Hs and Tp from the two networks on the shuffled measured data.



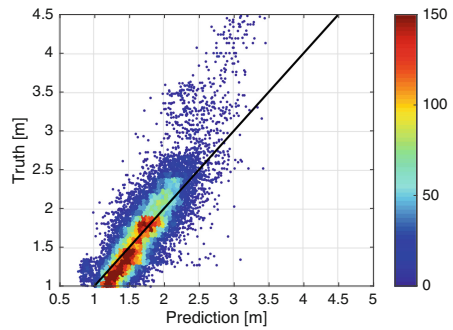
(a) D_p (MLSTM-CNN)



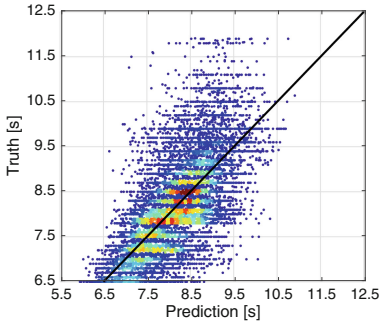
(b) D_p (Sliding Puzzle)



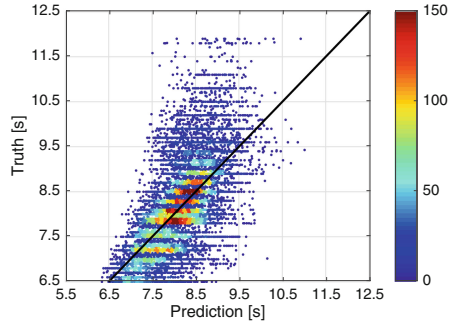
(c) H_s (MLSTM-CNN)



(d) H_s (Sliding Puzzle)



(e) T_p (MLSTM-CNN)



(f) T_p (Sliding Puzzle)

Fig. 11. Truth and prediction values for D_p , H_s and T_p from the two networks on the chronological measured data.

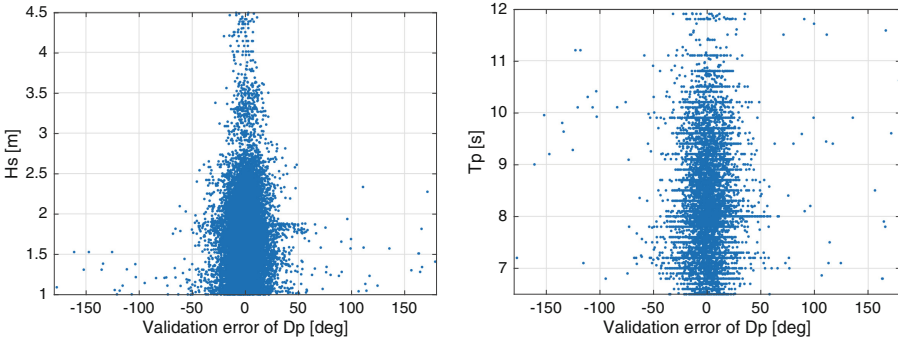


Fig. 12. Validation error of Dp versus Hs and Tp on the shuffled measured data using MLSTM-CNN.

The results show that the machine learning approach works, that there are some differences in performance between the two networks and that the way the data is used to train and evaluate the performance has quite some impact on the results.

To fully evaluate the results, these aspects will be discussed in the next three sections, together with a comparison to state of the art methods for sea state estimation.

Comparison to Spectral Sea State Estimation from Ship Motions

The model driven spectral methods use known response functions for mapping between sea state and ship motions. Although this approach works well for simulated data, it has some difficulty to find the relative wave direction for measured data. Our best methods seem to compare well to the sea trial results reported in [4, 15].

The main difference is the need for data. While the model driven method could work for any ship in any condition, the performance of our data driven method depends heavily on the availability of good quality measured data. However, this dependency may change in the future. Neural networks have the capability of generalization, meaning that, for this problem, data coming from multiple different ships can be used to train a general network that can be used with ships that were not in the training set. This idea is explored in [5] for simulated data.

Neural Network Structures

MLSTM-CNN and Sliding Puzzle perform similarly on both datasets. Even though MLSTM-CNN adds complexity in a parallel track, the global average pooling greatly reduces the complexity of the convolutional part. Furthermore, the parallel track supplies a different view on the data, counteracting overfitting on features as well. The Sliding Puzzle network also deliberately reduces the complexity by limiting the convolutional stage and collapsing filter results by

different forms of global pooling. This way both methods use a natural form of regularization, fitting the properties of the data¹.

Chronology of the Data

The measured data was collected over two years, in which the ship encountered many different sea states. Within the data set, the wave characteristics are estimated accurately though the performance for T_p is worse than for H_s and D_p . However, it is important to know how good the predictions will be when new data is seen. Since no samples have been duplicated, there is no contamination of the validation sets. Still there is a problem.

Keeping the data ordered chronologically before splitting into a training and validation set shows rather different validation results. Since this is a clear measure of the expected performance on new, unseen data, the previous case must have had some sort of contamination. Since we are using time series data for regression, we enter a realm in which the rules are different. Since the conditions the ship sailed in gradually change over time, the situation we have here is best described by a simple analogy. Suppose a network needs to be trained to classify different breeds of cats in images. The data set that is given contains an equal number of images for each breed, grouped per breed. If the network is trained on at least some images for every breed, it will be able to accurately classify any image from the validation set. If some breeds were absent in the training set, performance on the validation set will deteriorate. This will happen if the validation set consists of consecutive samples from the original chronological set. A common solution is to shuffle the dataset before splitting. Similarly, if we do not shuffle the ship motion/sea state dataset before splitting, some weather patterns will not be present in the training set, as they simply only occurred at specific times.

In a classification problem, the dataset is supposed to contain all classes. In our regression problem, with a dataset that grows over time, we do not have that guarantee. So, splitting before shuffling gives an estimate of the performance on new data. If the performance is significantly worse than for the shuffled set, the data set is not complete yet, turning this into an important measure. The score on the shuffled set is still a good indication of the possible performance on a complete set.

4.4 Results from the Numerical Simulations

Scatter plots of the truth and prediction values from the two networks are shown in Fig. 13, and the results for the evaluation metrics are listed in Table 2. The two networks exhibit an almost uniform performance in terms of prediction accuracy over the entire range of H_s , T_p and D_p . Between the two, MLSTM-CNN performs better for D_p , while Sliding Puzzle performs better for T_p . The networks perform

¹ Designing neural networks to general properties of the data is different from data snooping, as no sample knowledge is used, just knowledge about the way the data is acquired.

similarly for Hs. Similar to the performance on the measured data, the scatter around the truth values is larger for Tp compared to Hs and Dp.

Table 2. Results from the two neural networks applied to the simulation data.

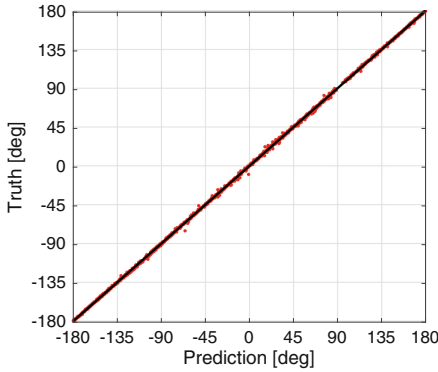
Train./Valid. data	Wave characteristic	95% level val.	Mean of val. error	Std of val. error	Network name
Simulation	Dp [deg]	1.86	0.02	0.98	MLSTM-CNN
		3.63	0.0	1.90	Sliding puzzle
	Hs [m]	0.51	0.0	0.24	MLSTM-CNN
		0.58	-0.01	0.27	Sliding puzzle
	Tp [s]	1.06	-0.02	0.51	MLSTM-CNN
		0.74	0.0	0.36	Sliding puzzle

4.5 Reusing the Models Trained on Simulated Data for the Measured Data

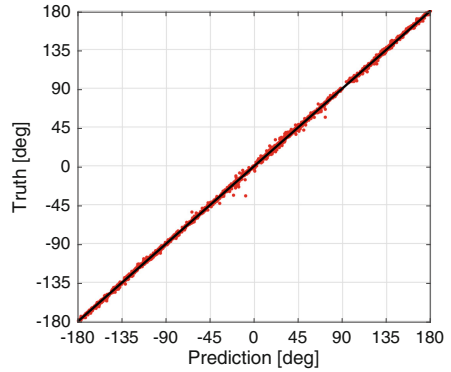
In this section, results are presented from the two networks that were trained on the simulated data and subsequently reused on the in-service measurement data. Two approaches were adopted when reusing the models on the measured data:

- **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.

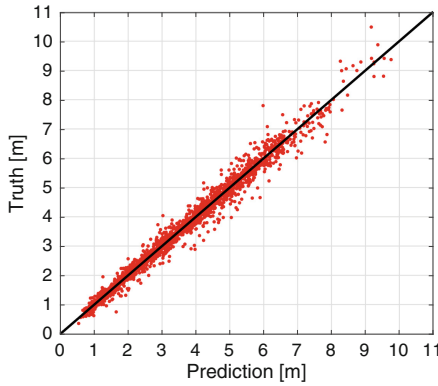
Figures 14 and 15 illustrate the results from the two networks when the direct application and transfer learning approaches are adopted, respectively. Table 3 lists the results for the evaluation metrics. It can be observed that the transfer learning approach results in a considerably better performance with both networks 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 decoded correctly. Between the neural networks, Sliding Puzzle performs less worse than the MLSTM-CNN with the direct application approach, while MLSTM-CNN performs better with the transfer learning approach. The 95% levels of validation errors for Dp of 30.62°, Hs of 0.42 m and Tp of 1.13 s are acceptable results opening the door for further studies with reusing the models trained on simulated data for the measured data.



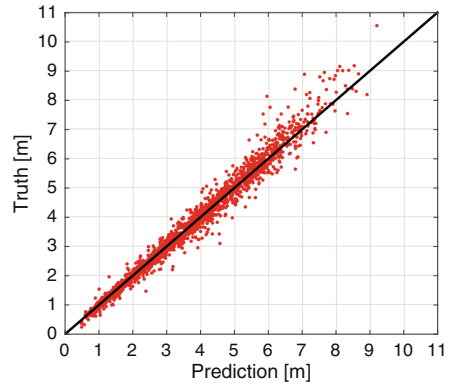
(a) D_p (MLSTM-CNN)



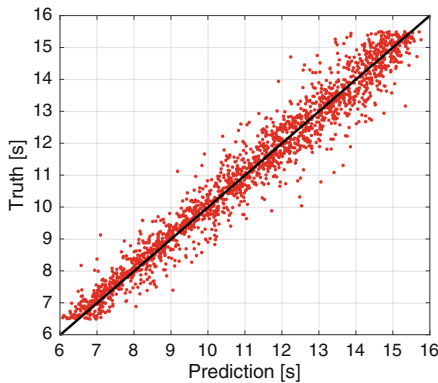
(b) D_p (Sliding Puzzle)



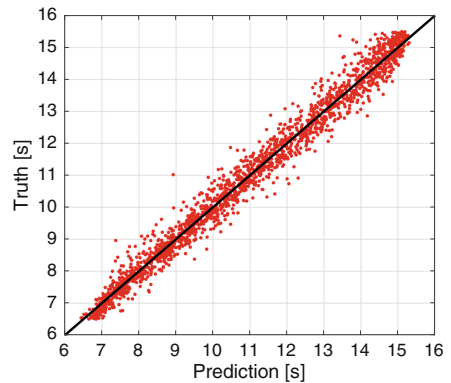
(c) H_s (MLSTM-CNN)



(d) H_s (Sliding Puzzle)



(e) T_p (MLSTM-CNN)



(f) T_p (Sliding Puzzle)

Fig. 13. Truth and prediction values for D_p , H_s and T_p from the two networks on the simulation data.

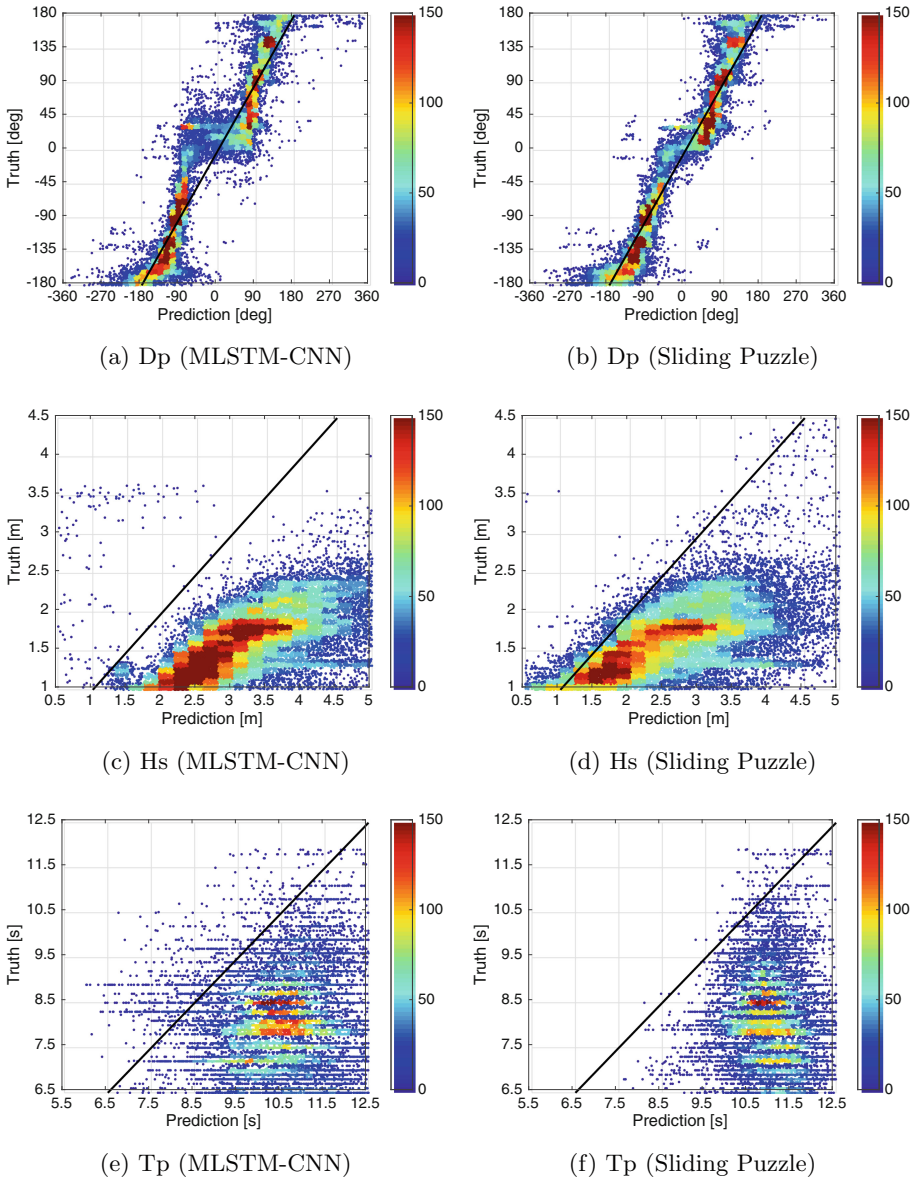
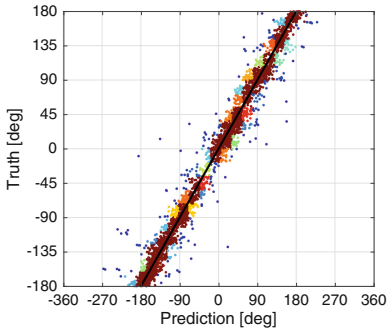
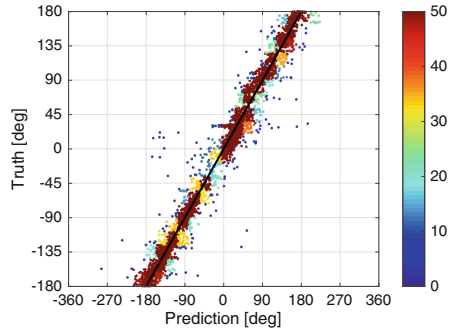


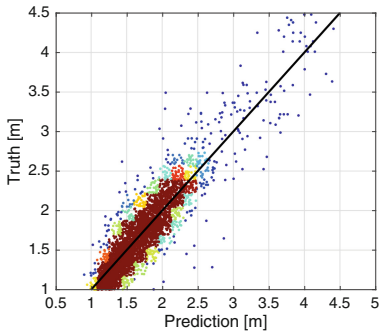
Fig. 14. Truth and prediction values for Dp, Hs and Tp from the two networks with the direct application approach.



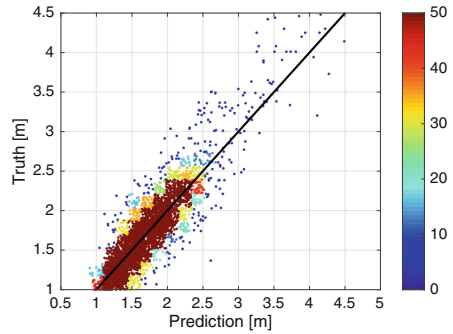
(a) D_p (MLSTM-CNN)



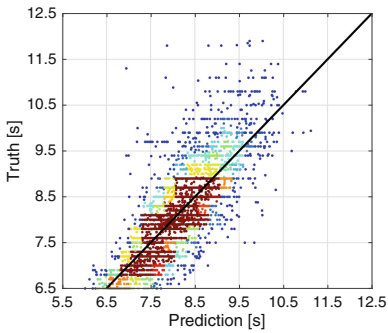
(b) D_p (Sliding Puzzle)



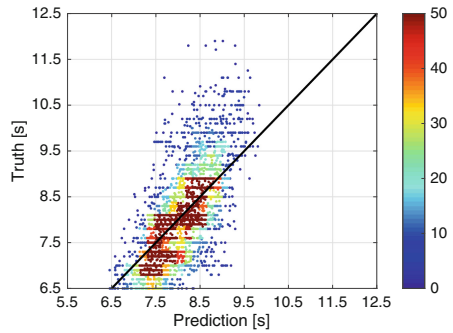
(c) H_s (MLSTM-CNN)



(d) H_s (Sliding Puzzle)



(e) T_p (MLSTM-CNN)



(f) T_p (Sliding Puzzle)

Fig. 15. Truth and prediction values for D_p , H_s and T_p from the two networks with the transfer learning approach.

Table 3. Results from the two neural networks with the direct application and transfer learning approaches.

Train./Valid. data	Wave characteristic	95% level val.	Mean of val. error	Std of val. error	Network name
Direct application	Dp [deg]	87.72	-4.91	40.46	MLSTM-CNN
		62.83	-1.86	31.60	Sliding puzzle
	Hs [m]	3.81	1.50	1.19	MLSTM-CNN
		2.70	1.0	0.91	Sliding puzzle
	Tp [s]	4.58	2.35	1.32	MLSTM-CNN
		4.73	2.85	1.16	Sliding puzzle
Transfer learning	Dp [deg]	30.62	-0.06	16.20	MLSTM-CNN
		34.23	0.02	18.32	Sliding puzzle
	Hs [m]	0.42	0.0	0.21	MLSTM-CNN
		0.46	0.0	0.22	Sliding puzzle
	Tp [s]	1.13	-0.01	0.57	MLSTM-CNN
		1.37	0.0	0.66	Sliding puzzle

5 Conclusions

In this study, machine learning approaches were employed to estimate wave characteristics from time histories of ship motions. For that purpose, both simulated and measured data were used to train the models. Two 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 was also investigated.

With the measurement data, the methods presented in this paper show good results and compare well to established methods. By including convolutional filters to encode the phase relations between the 6-DOF motions, the adopted neural networks were capable to estimate the wave characteristics accurately especially for the significant wave height and main direction. The data used to achieve this does show some shortcomings though and does not cover the full spectrum. More good quality data is needed to improve methods based on measured data.

The way the data is used for training is crucial when evaluating method performance. We have seen that there is quite a difference in performance between using the data chronologically and shuffled. Even though shuffling is common practise, it is to be used carefully when using data acquired over time. The advice is to use both training strategies to get insight in both the possible performance and the actual performance when applied to new data.

With the simulation data, the results indicate that the neural networks show a very good performance in general. Similar to the results on the measured data, the performance of the networks was better for Hs and Dp compared to Tp.

Finally, the neural networks trained on the simulated data were reused on the in-service measurement data. The results show that the learned features of the models 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 the wave characteristics accurately. However, this accuracy level was lower than the one obtained when the neural networks were directly trained on the measured data itself.

Future work will include more complex numerical simulation studies with short-crested waves and combined wind and swell sea spectrums. Model test data will also be utilized for validation. Furthermore, the neural networks will be trained to estimate the full directional spectrum of the waves (encoder-decoder networks).

References

1. Tannuri, E.A., Sparano, J.V., Simos, A.N., Cruz, J.J.D.: Estimating directional wave spectrum based on stationary ship motion measurements. *Appl. Ocean Res.* **25**, 243–261 (2003)
2. Iseki, T., Ohtsu, K.: Bayesian estimation of directional wave spectra based on ship motions. *Control Eng. Pract.* **8**, 215–219 (2000)
3. Nielsen, U.: Estimations of on-site directional wave spectra from measured ship responses. *Marine Struct.* **19**, 33–69 (2006)
4. Nielsen, U.D., Brodtkorb, A.H., Sørensen, A.J.: A brute-force spectral approach for wave estimation using measured vessel motions. *Marine Struct.* **60**, 101–121 (2018)
5. Mak, B., Düz, B.: Ship as a wave buoy - using simulated data to train neural networks for real time estimation of relative wave direction. In: *Proceedings of the ASME 2019 38th International Conference on Ocean, Offshore and Arctic Engineering OMAE2019*, paper OMAE2019-96225, Glasgow, Scotland (2019)
6. Mak, B., Düz, B.: Ship as a wave buoy - estimating relative wave direction from in-service ship motion measurements using machine learning. In: *Proceedings of the ASME 2019 38th International Conference on Ocean, Offshore and Arctic Engineering OMAE2019*, paper OMAE2019-96201, Glasgow, Scotland (2019)
7. Karim, F., Majumdar, S., Darabi, H., Harford, S.: Multivariate LSTM-FCNs for time series classification. *arXiv 1801.04503v1* (2018)
8. McTaggart, K., De Kat, J.O.: Capsize risk of intact frigates in irregular seas. *Trans. SNAME* **108**, 147–177 (2000)
9. Van't Veer, R., De Kat, J.O.: Experimental and numerical investigation on progressive flooding and sloshing in complex compartment geometries. In: *Proceedings of the 7th International Conference on Stability for Ships and Ocean Vehicles (STAB 2000)*, February, Launceston, Australia, vol. A, pp. 305–321 (2000)
10. Stredulinsky, D.C., Thornhill, E.M.: Ship motion and wave radar data fusion for shipboard wave measurement. *J. Ship Res.* **55**, 73–85 (2011)
11. Kiefer, J., Wolfowitz, J.: Stochastic estimation of the maximum of a regression function. *Ann. Math. Stat.* **23**(3), 462–466 (1952)
12. Chollet, F., et al.: Keras (2015). <https://github.com/fchollet/keras>
13. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). <https://www.tensorflow.org/>. (Software available from tensorflow.org)

14. Intel.ai: PlaidML (2018). <https://www.intel.ai/plaidml>
15. Bispo, I., Filho, A., Tannuri, E., Simos, A.: Motion-based wave inference: monitoring campaign on a turret FPSO. In: Proceedings of the ASME 2016 35th International Conference on Ocean, Offshore and Arctic Engineering OMAE2016, paper OMAE2016-54956, Busan, South Korea (2016)