Modeling and Analysis of Motion Data from Dynamically Positioned Vessels for Sea State Estimation

Xu Cheng[1,2], Guoyuan Li[2*], Robert Skulstad[2], Shengyong Chen[1], Hans Petter Hildre[2], and Houxiang Zhang[2]

***Abstract*— Developing a reliable model to identify the sea state is significant for the autonomous ship. This paper introduces a novel deep neural network model (SeaStateNet) to estimate the sea state based on the ship motion data from dynamically positioned vessels. The SeaStateNet mainly consists of three components: an Long-Short-Term Memory (LSTM) recurrent neural network to capture the long dependency in the ship motion data; a convolutional neural network (CNN) to extract time-invariant features; and a Fast Fourier Transform (FFT) block to extract frequency features. A feature fusion layer is designed to learn the degree affected by each component. The proposed model is applied directly to the raw time series data, without needing of any hand-engineered features. A sensitivity analysis (SA) method is applied to assess the influence of data preprocessing. Through benchmark test and experiment on ship motion dataset, SeaStateNet is verified effective for sea state estimation. The investigation on real-time test further shows the practicality of the proposed model.**

I. INTRODUCTION

As the complex marine operations have been moving towards the ultra-deep sea, the demanding of new technologies and equipment is increasing to make the operations more safe for the harsh environment [1]. The working window of vessels is weather-dependent, which requires adequate understanding of the weather condition to reduce cost and improve safety [2]. Recently, there is a trend to consider developing more advanced vessels that have intelligence and are capable of executing different levels of autonomy for maritime operations. Developing a real-time and reliable model to estimate the sea state is significant to aid the decision making for the autonomous ship.

Traditional instruments such as wave buoys, X-band radars and remote sensing satellites can estimate the sea state to some extent, but have their own limitations. The wave buoys can work close to the shore, but need to be placed for every measurement, and it is hard to apply it to predict the sea state in the open sea. The X-band radars need to be calibrated frequently, and the cost of installation and use is high. The remote sensing satellites is easily affected by the cloud, and the weather information just lags up to several hours, in general. Nowadays, many researchers have conducted extensive explorations on the identification of sea state based on the ship motion data directly [3], [4]. A ship can be considered as a big wave buoy, thus, inherently equipped with a sea state estimation system [2].

The use of external sensors (wave buoys, etc.) or the ship's own motion data to identify sea states usually involves the analysis in the time domain or in the frequency domain [5]. The idea of sea state estimation in time domain is updating the sea state from the ship motion measurement. While the estimation of sea state in frequency domain would usually combine the response spectrum with the response amplitude operators (RAOs). The drawback of those method is that both methods highly rely on mathematical models or assumptions in either time domain or frequency domain, and the recent collected data may easily lead to incorrect identification due to the randomness of the environmental factors. To address the problem of conventional methods, other researchers turned their attention to machine learning, using ship motion data and feature engineering techniques to extract temporal and frequency domain features from the data. For example, in [6], the sea state was estimated by a multi-layer random forest (RF) classifier. The method does not rely on accurate mathematical models, but requires many hand-crafted features that play a very important role in the classification results.

To the best of our knowledge, even though deep learning (DL) based classification have been hot topic in recent years, the study on exploring DL in sea state estimation is seldom seen. Applying the DL to the sea state estimation based on a DP ship's motion data has several challenges: 1) the DP ship motion data contains the information of the external environmental factors and the control forces from the thruster system. Thus, it is not easy to learn the environmental information purely based on the data. 2) Changeable weather in the open sea brings a lot of transitions of sea state. The uncertainty of sea state transition brings difficulties to the estimation of sea state. 3) The status of DP ship in different sea state could be very similar, which makes it very hard to determine the sea state if just focusing on current motion data. 4) The sensor data contains measurement-induced noise and uncertainties, which brings difficulties to modeling.

To address the above challenges, a novel end-to-end deep neural network called SeaStateNet is proposed to accurately classify the sea state based on ship motion data. The SeaStateNet directly utilizes the raw sensor data for modeling, in which a SA method is employed to assess and calibrate the influence of data processing to the classification results. We have made the following main contributions in this work.

- An end-to-end DL network (SeaStateNet) is designed, combining LSTM, CNN, FFT with feature fusion is

*Corresponding author

Xu Cheng and Robert Skulstad have equal contribution.

[1]Xu Cheng and Shengyong Chen are with the School of Computer Science and Technology, Tianjin University of Technology, Tianjin, 300384, China.

[2]Xu Cheng, Guoyuan Li, Robert Skulstad, Hans Petter Hildre, Houxiang Zhang are with the Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Aalesund, 6009 Norway.

proposed for sea state estimation.
- The proposed model is evaluated on the publicly benchmark datasets, and ship motion dataset from a commercial simulator, together with real-time use of the model in the simulator.

The paper is organized as follows: an introduction to sea state estimation and time series classification technologies is given in Section II. Section III presents the design of network. The experiments are discussed in Section IV. Section V concludes the paper.

II. Related Work

A. Sea State Estimation

Sea state is defined as a general condition in a certain location and moment, which is characterized by the wave and wind. Table I describes the sea code which is proposed to define the sea state quantifiably. Most of the research for sea state estimation recently is combining the wave-induced response measurements, which are collected by wave buoys or directly from the ship motion, and a mathematical model. The previous work focused on the field of frequency domain analysis where the wave (energy) spectrum would be given [7], [8], [9], [10], [11], [2]. Pascoal and Soares [12] proposed a Kalman filtering based method which relies on the accurate RAOs in time domain only. Nielsen et al. [13] also calculated the sea state directly in time domain on the basis of the measured response and corresponding theory. It should be mentioned that all the above methods ship dependent, which means their methods can only be used for these specific vessels. Nevertheless, from [6], if the model depends only on ship motion data, it could be applicable to all types of vessels. In essence, ship motion data is time series data, and contains several frequencies which can describe the characteristics of environment. Thus, it is promising to establish a deep learning model to identify the sea state which can extract information from the time and frequency domain, and doesn't need to rely on hand-craft features.

TABLE I: Code of sea state [14]

Sea State Code	Description of sea	Wave height observed (m)	World wide probability (%)
0	Calm (glassy)	0	-
1	Calm (ripples)	0-0.1	11.2486
2	Smooth	0.1-0.5	-
3	Slight	0.5-1.25	31.6851
4	Moderate	1.25-2.5	40.1944
5	Rough	2.5-4.0	12.8005
6	Very rough	4.0-6.0	3.0253
7	High	6.0-9.0	0.9263
8	Very High	9.0-14.0	0.1190
9	Extreme	Over 14.0	0.0009

B. Time Series Classification

The algorithm of time series classification have been developed over years. The distance-based methods for time series classification have been the first successful algorithm [15]. Feature-based algorithms also are widely used. Generalized Random Shapelet Forests (gRSF) [16], and Hidden Unit Logistic Model (HULM) [17] are two successful feature-based algorithms that have obtained the state-of-the-art results on various benchmark datasets. Some approaches also employ the dimensionality reduction techniques for time series classification. Learned Pattern Similarity (LPS) [18] can be used to extract information from the multivariate time series to train a regression tree to find underlying dependencies. A hybrid multivariate time series classification method, named as WEASEL+MUSE, proposed by Schäfer, and Leser [19], adopts the idea of bag-of-pattern approach and achieved high competitive accuracies. Deep learning has also been applied to the multivariate time series classification. Zheng et al. proposed a multi-channel deep convolutional neural network (MCDCNN) [20] to utilize the features learned from different channels for multivariate time series classification. An end-to-end LSTM-FCNs model employ the parallel LSTM and FCN to extract the feature, and achieved state-of-the-art results on various benchmark datasets [21]. In summary, it is a good choice to make use of the advantage of deep learning and combine with the inherent characteristics of ship motion for sea state estimation.

III. Proposed approach

A. Structure

A considerable amount of sensor data can be gathered during DP operation. How to utilize these data to establish model for sea state estimation has the following challenges: 1) the sensor data to be applied are usually too large and high dimensional, and there are many uncertainties in these data which are influenced by various factors, such as weather conditions and human factors; 2) sensor data is collected in time order, increasing the difficulty of obtaining the dynamics of the underlying process; 3) How to properly process the raw data not only improve the estimation accuracy of the model but also find a reasonable estimation interval to cope with environmental changes is challenging.

To address the above challenges, a structure is proposed, as shown in Fig. 1. To capture more information of time series data, the proposed model is to combine LSTM block, CNN block, and FFT block to utilize their representation abilities on different aspects of data. To characterize and reduce that uncertainty and find the reasonable estimation interval in the sensor data, SA would be utilized [22]. Technically, LSTM block is training over the sequence of the sensor data to capture the dependency in historical sensor data. The CNN block is utilized to extract features from the sets of local data. In order to make use of the prior knowledge in the field of sea state, FFT block is also used to help to learn the feature in frequency domain. Then, the feature fusion layer merges the features for classifying the sea state. Finally, the sea state classification is realized by the MLP. It is worth noting that the the three parallel parts consist of H stacked layers and H can be different for each layer. As illustrated in Fig. 1, H_1 and H_2 stand for the first layer and the second layer in each block.

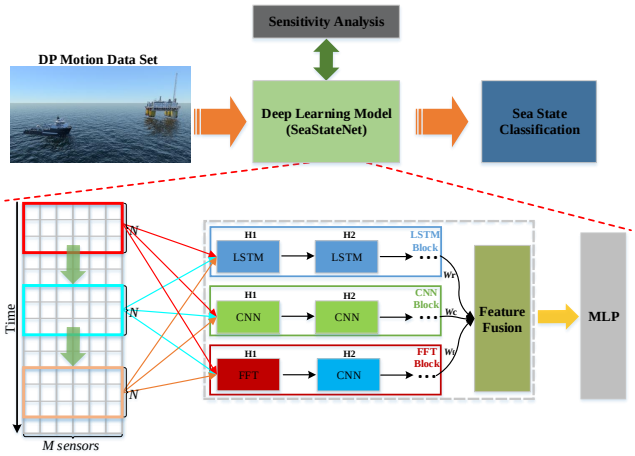
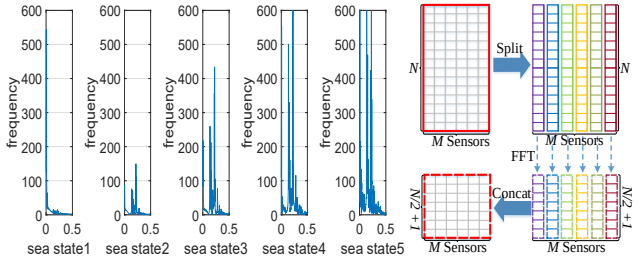


Fig. 1: Illustration of the proposed Structure.



(a) Frequency of different sea state. (b) FFT of time series data.

Fig. 2: Performance comparison of different variants.

B. Data Preprocessing

Time series is defined as a vector $X = \{x^{(1)}, x^{(2)}, ..., x^{(t)}, ..., x^{(T)}\}$, where T is the length of time series, and each element $x^{(t)}$ contains M values, e.g. $x^{(t)} = \{x_1^{(t)}, x_2^{(t)}, ..., x_M^{(t)}\}$, representing M number of sensors.

The task of time series classification is to estimate the label $Y = \{1, 2, ..., C\}$, and the time series model can be expressed as: $y = f(X)$, $y \in Y$, which can be learned from the data. Since the LSTM usually fails to capture the very long-term correlation [23], the long raw time series should be truncated into small segments. As seen from Fig. 1, the length of each segment is N. In practical applications, there is no guarantee that each small segment will have the same length. Therefore, when the length is less than N, the sequence is padded with zeros.

C. Sensitivity Analysis

In this paper, we consider that several factors that influence the performance of the proposed model: First, how to select the input parameters that are most relevant to the output for model construction; Second, there are many hyper-parameters that need to be calibrated. Third, how to process the raw time series data also would affect the model outputs.

In this paper, the mutual information (MI) based SA is employed to select those influential sensor data [24]. The MI based SA is a data-based approach, which is efficient and effective to use, for it can learn the dependence between variables from the sensor data directly. In addition, empirical SA methods [25], which are widely used in the analysis of DL model, are used here to determine the impact of data pre-processing. Details please see Section IV-E.

D. Learning the Temporal Dependency

During the training phase, each small segment in the long-term time series data is fed into LSTM block. The k-th LSTM unit consists of a memory cell C_k, which is composed of four components: input gate, forget gate, output gate, and cell state, that can store information for a long time by updating the internal state. The output of this LSTM unit can be expressed as: $h_k = O_k tanh(C_k)$.

O_k is calculated by

$$\begin{aligned} O_k &= \sigma(\boldsymbol{W}_{xo} x_k + \boldsymbol{W}_{ho} h_{k-1} + \boldsymbol{W}_{co} C_k + b_o) \\ C_k &= f_k C_{k-1} + i_k tanh(\boldsymbol{W}_{xc} x_k + \boldsymbol{W}_{hc} h_{k-1} + b_c) \\ f_k &= \sigma(\boldsymbol{W}_{xf} x_k + \boldsymbol{W}_{hf} h_{k-1} + \boldsymbol{W}_{cf} C_{k-1} + b_f) \\ i_k &= \sigma(\boldsymbol{W}_{xi} x_k + \boldsymbol{W}_{hi} h_{k-1} + \boldsymbol{W}_{ci} C_{k-1} + b_i) \end{aligned} \tag{1}$$

where x is the input vector to the LSTM unit; i, f, and o are activation vectors used, to control the forget gate, the input gate, and the output gate; C represents the cell input activation vector, respectively; and h is the cell output vector. The function σ is the logistic sigmoid function, and $\boldsymbol{W}$ and b represent weight matrices and bias associated with different activation vectors of each equation. In this paper, the LSTM block contains a dropout layer to avoid the over-fitting.

E. Learning the Local Features from Time Series Data

When the k-th data block is fed to LSTM, the corresponding local raw time series data point is also input to CNN block of SeaStateNet. CNN block consists of H stacked layers of 1D convolutional units. In general, each 1D convolutional unit contains a convolutional layer that is accompanied by batch normalization. After batch normalization is the activation function of the rectified linear unit. Each convolutional layer has a specified number of filters with a specified filter size. Each filter on the layer scans the entire data block to extract local features.

F. FFT Layer

Fig. 2a shows the spectrum of the roll of ship under different sea states. The distinction between the amplitude of different sea states is quite obvious. And there is frequency offset for different sea states. This means that the use of frequency domain information for the classification of sea state has significant benefits.

The process of FFT is shown in Fig. 2b. The k-th data segment with shape $(N \times M)$ would be split to M $(N \times 1)$ matrices. Then, the FFT would be performed for each $(N \times 1)$ matrix. For the reason of FFT algorithm, the length of $(N \times 1)$ matrix would reduced to $N/2+1$. The M $(N/2+1 \times 1)$ FFT matrices would be concatenated to one block.

1D convolution layer is utilized to capture the local features from the frequency data. The setting of 1D convolution layer is the same with the CNN block.

TABLE II: Accuracy comparison with the state-of-art time series classification methods

DataSet	DTW	gRSF	MLSTM -FCN	WEASEL + MUSE	SeaStateNet
ArabicDigits	0.908	0.975	**1.000**	0.992	0.994
AUSLAN	0.727	0.955	0.970	**0.991**	0.978
CharTrajectories	0.948	0.994	**1.000**	0.973	0.994
CMUsubject16	0.930	**1.000**	**1.000**	**1.000**	**1.000**
ECG	0.790	**0.880**	0.860	**0.880**	0.860
JapaneseVowels	0.962	0.8	**1.000**	0.976	0.995
KickvsPunch	0.600	**1.000**	**1.000**	**1.000**	**1.000**
Libras	0.888	0.911	0.970	0.894	**0.981**
NetFlow	**0.976**	0.914	0.950	0.938	0.933
UWave	0.916	0.929	**0.980**	0.916	**0.980**
Wafer	0.974	0.992	0.990	**0.997**	0.992
WalkvsRun	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Robot Failure LP1	0.760	0.840	0.860	**0.940**	0.860
Robot Failure LP2	0.700	0.667	0.830	0.733	**0.833**
Robot Failure LP3	0.567	0.633	0.800	**0.900**	0.800
Robot Failure LP4	0.867	0.867	0.920	0.960	**0.987**
Robot Failure LP5	0.540	0.450	0.660	0.690	**0.710**
PenDigits	0.927	0.932	**0.970**	0.912	0.967
Shapes	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
DigitShapes	0.938	**1.000**	**1.000**	**1.000**	**1.000**
Avg. accuracy	0.846	0.887	0.938	0.935	**0.943**
Wins/Ties	3	6	**10**	**10**	**10**

G. Feature Fusion Layer and MLP

The feature fusion layer combines the output representations from LSTM, CNN and FFT. Then, the joint feature is fed to the MLP to provide the sea state classification. Assume that these outputs can be represented by LSTM(X), CNN(X), and FFT(X). Particularly, they would be mapped to the same feature space and then add them together to obtain the activation of the feature fusion layer. The three components can be fused as follows:

$$X_{fusion} = \boldsymbol{W}_r \cdot LSTM(X) + \boldsymbol{W}_c \cdot CNN(X) + \boldsymbol{W}_f \cdot FFT(X) \quad (2)$$

where $\cdot$ is element-wise multiplication. $\boldsymbol{W}_r, \boldsymbol{W}_c, \boldsymbol{W}_f$ are the learnable parameters that adjust the degrees affected by LSTM(X), CNN(X), and FFT(X), respectively. The output layer, which uses Softmax as the activation function, is a fully-connected layer following the feature fusion layer.

IV. Experiments

All experiments were performed on a server equipped with an Intel Xeon processor, 128 GB RAM and Nvidia Tesla K80 and 24 GB RAM. The software environment used is Anaconda[1] Python 3.6, and all the layers are implemented by Keras 2[2], using TensorFlow[3] as the backend.

[1]https://anaconda.org/

[2]https://keras.io

[3]https://www.tensorflow.org/

A. Datasets

- **Benchmark dataset**: We evaluated our proposed model SeaStateNet using 20 publicly available MTS datasets collected in [21], [19]. SeaStateNet is compared to five state-of-art time series classification methods, using results reported by their respective authors in their publications.
- **Ship motion dataset**: The ship motion data comes from the Offshore Simulator Centre AS (OSC) [26]. According to Table I, the world wide probability of sea state 0 - 5 occurring in the world accounts for almost 96%. Thus, we simulated the six sea states from sea state 0 to sea state 5. Since the first two sea states are very similar, they are merged to one state. In order to reflect the complexity of the change of environment, waves and winds are randomly generated no greater than the range of sea state 5, and the transition also random happens per [5, 30] minutes. In this experiment, the ship motion data (roll, yaw, pitch, surge velocity, sway velocity, heave velocity, roll velocity, pitch velocity and yaw velocity) is used. we partition the data into non-overlapped training, test data by 80% and 20%.

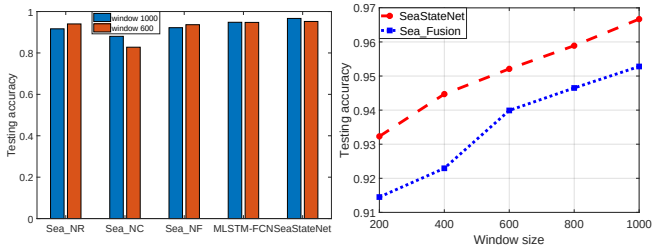
B. Comparison With the State-of-art Time Series Classification Methods

To illustrate the feasibility of our proposed model, our model is first compared with the state-of-art methods. In these tests, the stacked layers H for LSTM, CNN, and FFT block are set to $\{2,3,2\}$. The mini-batch size is set to 128, and the number of units of LSTM is set to 8 and 8. The number of filter of *CNN* block is 128, 256 and 128 with kernel size 3, 5 and 3. The number of filter and kernel size for FFT block are 128 and 5, respectively. Adam optimizer [27], with an initial learning rate set to 1e-3 and the final learning rate set to 1e-4, is employed to train all models. Each benchmark dataset offers training and testing split which we make use to compare to the prior publications. The window size of each dataset of our method is the same as [21].

Table II summarizes the test results of the time series classification method. From Table II, SeaStateNet achieved highest average accuracy compared to the other approaches, and has similar ranking (Wins/Ties) with WEASEL+MUSE and MLSTM-FCN. Our model is able to achieve good results on most data sets. In particular, on a few data sets, such as LP4, LP5, our model achieved better results. The reason behind may be that these data sets have more distinguishable features in the frequency domain. Compared with MLSTM-FCN, our model has one more FFT part. The next section will show that the additional FFT layer can improve the accuracy for sea state estimation.

C. Comparison with Baselines

We compared SeaStatNet with LSTM, CNN, and MLSTM-FCN on the ship motion data set. The reason for comparing these methods is that these methods belong to the concept of end-to-end deep learning and does not rely on data pre-processing. We did not use the WEASEL+MUSE



(a) (b)

Fig. 3: Performance comparison of different variants.

method mentioned in Section IV-B because the method heavily depends on the hand-craft features. To evaluate the performance of each model, those models are tested on datasets with different window size N ranging from 200 to 1000. In this test, the number of hidden units of LSTM is set to $\{8,16,32,64,128\}$, the number of filters of CNN is set to $\{64,128,256,512\}$. The setting of MLSTM-FCN is the same to [21]. The H of SeaStateNet is set to $\{1,2,2\}$. The mini-batch size is set to 256, and the number of units of LSTM is set to 8. The number of filter of CNN block is 128 and 256 with kernel size 3 and 5. The number of filter and kernel size for FFT block are 128 and 5, respectively. To be fair, we choose the best LSTM, CNN and MLSTM-FCN to compared with our model.

Table III reports the classification accuracy of the four models, evaluated on the testing set. As can be clearly seen from the table, the performance of LSTM is the worst compared to the other three methods. As the length of the window increases, the over-fitting of the LSTM becomes more serious. CNN is better than the LSTM (almost 11% improvement). SeaStateNet achieved the best results on all testing datasets. Compared with MLSTM-FCN, and with the increase of window length, the accuracy of classification increases gradually. The model structure proposed in this paper uses one CNN layer less than MLSTM-FCN, but has one more FFT layer. The reason for this is that the FFT block can get more frequency domain information with the growth of window size.

TABLE III: Accuracy comparison with baselines on ship motion data.

Model / Window size	**LSTM**	**CNN**	**MLSTM-FCN**	**SeaStateNet**
1000	0.7909	0.8917	0.9435	0.9667
800	0.8237	0.9011	0.9481	0.9589
600	0.8022	0.8923	0.9475	0.9521
400	0.7867	0.8710	0.9274	0.9447
200	0.8300	0.8662	0.9226	0.9323
Average	0.8067	0.8845	0.9378	0.9509

D. Variant Comparison

To further investigate the effectiveness of component in the model, we compare SeaStateNet with its variants as follows:

- **Sea_NR**: There is no LSTM in the SeaStateNet.
- **Sea_NC**: To validate the effects of CNN block, we remove it from SeaStateNet.
- **Sea_NF**: This variant does not consider the effects of FFT block.
- **Sea_Fusion**: The feature fusion layer is removed from SeaStateNet.

Each variant was tested based on the same ship motion dataset, with $N = 600$ and $N = 1000$, respectively. The settings of hyper-parameter of SeaStateNet is the same with Section IV-C. The total epoch number is set to 600. The experimental results are presented in Fig. 3. From the Fig. 3a, we observe that: 1) the full combination of the four components shows superiority against Sea_NR,Sea_NC, and Sea_NF, which demonstrates the importance of the proposed model. 2) The biggest accuracy drop happens when CNN block is removed. This means the most important part is the CNN, rather than the LSTM or FFT.

To illustrate the importance of fusion layer, we further compare the network with/without the fusion layer. In the case of without fusion layer, direct method to fuse the three components, i.e., LSTM(X) + CNN(X) + FFT(X) is used. As depicted in Fig. 3b, the fusion layer can improve the accuracy by considering the importance of each component.

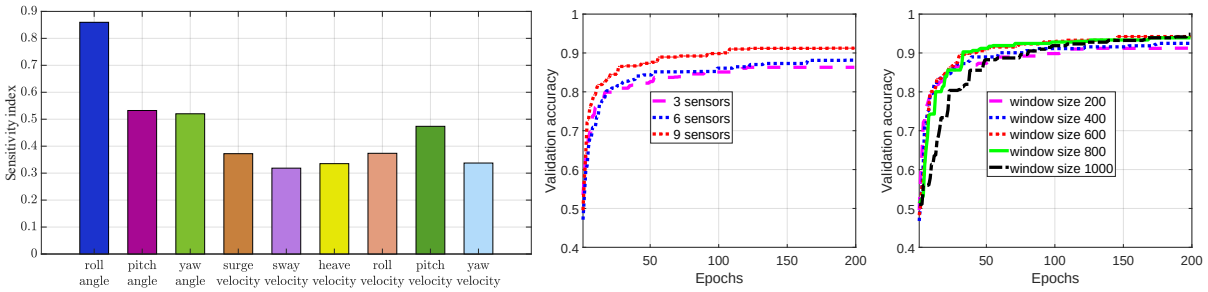
E. Sensitivity of Data Pre-processing

We study the sensitivity of SeaStateNet with respect to data pre-processing, i.e., the window size N, and the number of sensors M. In this paper, the raw time series contains nine sensors. To illustrate the importance of each sensor, MI is performed and the ranking-based MI index can be obtained, as shown in Fig. 4a. We take the 3 and 6 most important sensor inputs and compared them with full sensor inputs for $N = 1000$. As illustrated in Fig. 4b, the higher accuracy is achieved when the number of sensors is nine. More interestingly, the more sensors there are, the faster the convergence rate is. The explanation for this result is that the more sensors there are, the more information can be provided by the data.

Fig. 4c shows how the accuracy varies with respect to the change of window size when $M = 9$. It is obvious to observe that the convergence rates for window size $200,400,600,800$ are almost the same. However, the convergence rate for window size 1000 is slower at the beginning. Another finding is that the bigger window size, the higher accuracy becomes. This is reasonable because a bigger window size brings more solid and adequate information about the current sea state so that the estimation accuracy increases. The phenomenon will be more obvious when using short window size for estimating high sea state (such as sea state 5 in this paper).

F. Real-time Estimation

To further investigate our model, the proposed model is applied on the OSC simulator for real-time sea state estimation. The schematic of this test is shown in Fig. 5. The SeaStateNet performs real-time sea state estimation after receiving the data, and the result is sent back to the OSC



(a) MI of time series data. (b) Influence of sensors (c) Influence of window size

Fig. 4: Sensitivity analysis

platform for on-board support. Not only would it be helpful for the pilot to better understand the environment, but also promising if it is further integrated into the control system.

Two periods of the online test were recorded for performance analysis. The length of the first and second period are approximately 36 minutes and 1.5 hours, respectively. During the entire test, the OSC simulator produced data for six sea states, and the length of each state is also randomly generated from 5 minutes to 30 minutes. The model with window size 1000 is tested. The statistics of this test are shown in the Table IV. It is worth noting that the system does not produce sea state 3 in the first period of the test.

From Table IV, we can see that the bigger probability of sea state would bring larger error in both tests. This means that the transition of sea state would bring more errors. In our tests, the error comes from two major parts: the estimation error of the model itself and the error caused by the sea state transition. The estimated error for each state may be almost the same if the testing time is long enough. Thus, the majority of estimation errors come from the transitions. In the first period of the test, the order of occurrence of sea states is 1-5-2-5-4. Most of the transitions take place from 5 - 2 - 5, thus these two sea states (sea state 2 and sea state 5) contribute most of the errors. This is consistent to the analysis on the second period of the test, where the most frequently occurring sea states (sea state 4 and 5) correspond to a relative lower accuracy.

TABLE IV: Results of real-time tests

Sea State	**Probability of sea state**		**Test 1**	**Test 2**
	Test1	Test 2		
1	20%	12.5%	100%	93.47%
2	20%	12.5%	96.25%	88.81%
3	0%	6.25%	—	98.89%
4	20%	37.5%	99.74%	83.25%
5	40%	31.25%	89.86%	84.23%
Transition number	4	13	—	—
Average accuracy	—		**96.46%**	**89.73%**

V. CONCLUSIONS

In this paper, we propose a novel neural network based model to classify the sea state based on the ship motion data. In the model, an LSTM recurrent neural network is employed to capture the long dependency in the ship motion data, a CNN is utilized to extract time-invariant features, and a FFT is employed to extract frequency features. A feature fusion layer is implemented to adjust the relative influence of each of the three components. The proposed model is applied directly to the raw time series data and only needs slight data pre-processing, without needing any hand-engineered features. The proposed model is evaluated in the benchmark data-sets and ship motion dataset. The experiment show our model outperforms the baseline methods. The real tests demonstrate the practicality of our method.

In the future, we will extend our method to solve the problem of the sea state transition. Moreover, we will integrate the sea state estimation model to the ship motion monitoring system to serve the autonomous ship.

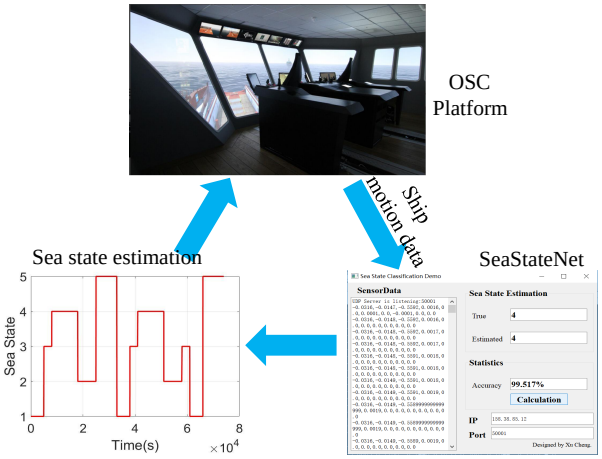


Fig. 5: Schematic of real-world application.

ACKNOWLEDGMENT

This work was supported in part by the project "SFI MOVE" and "Digital Twins for Vessel Life Cycle Service" (Project no.: 237929 and 280703) and in part by the National Natural Science Foundation of China (U1509207). The author Xu Cheng is financially supported by the Chinese Scholarship Council. The authors would like to thank the technical support from Offshore Simulator Centre AS.

REFERENCES

[1] A. J. Sørensen, "A survey of dynamic positioning control systems," *Annual reviews in control*, vol. 35, no. 1, pp. 123–136, 2011.

[2] A. H. Brodtkorb, U. D. Nielsen, and A. J. Sørensen, "Sea state estimation using vessel response in dynamic positioning," *Applied Ocean Research*, vol. 70, pp. 76–86, 2018.

[3] M. Loueipour, M. Keshmiri, M. Danesh, and M. Mojiri, "Wave filtering and state estimation in dynamic positioning of marine vessels using position measurement," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3253–3261, 2015.

[4] U. D. Nielsen, A. H. Brodtkorb, and A. J. Sørensen, "A brute-force spectral approach for wave estimation using measured vessel responses," *Marine Structures*, 2017.

[5] U. D. Nielsen, "A concise account of techniques available for ship-board sea state estimation," *Ocean Engineering*, vol. 129, pp. 352–362, 2017.

[6] F. Tu, S. S. Ge, Y. S. Choo, and C. C. Hang, "Sea state identification based on vessel motion response learning via multi-layer classifiers," *Ocean Engineering*, vol. 147, pp. 318–332, 2018.

[7] J. Hua and M. Palmquist, "Wave estimation through ship motion measurement," *Strip Method*, 1994.

[8] T. Iseki and K. Ohtsu, "Bayesian estimation of directional wave spectra based on ship motions," *Control Engineering Practice*, vol. 8, no. 2, pp. 215–219, 2000.

[9] U. D. Nielsen, "Estimations of on-site directional wave spectra from measured ship responses," *Marine Structures*, vol. 19, no. 1, pp. 33–69, 2006.

[10] R. Pascoal, C. G. Soares, and A. Sørensen, "Ocean wave spectral estimation using vessel wave frequency motions," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 129, no. 2, pp. 90–96, 2007.

[11] R. Pascoal, L. P. Perera, and C. G. Soares, "Estimation of directional sea spectra from ship motions in sea trials," *Ocean Engineering*, vol. 132, pp. 126–137, 2017.

[12] R. Pascoal and C. G. Soares, "Kalman filtering of vessel motions for ocean wave directional spectrum estimation," *Ocean Engineering*, vol. 36, no. 6-7, pp. 477–488, 2009.

[13] U. D. Nielsen, R. Galeazzi, and A. H. Brodtkorb, "Evaluation of ship-board wave estimation techniques through model-scale experiments," in *OCEANS 2016-Shanghai*. IEEE, 2016, pp. 1–8.

[14] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[15] C. Orsenigo and C. Vercellis, "Combining discrete svm and fixed cardinality warping distances for multivariate time series classification," *Pattern Recognition*, vol. 43, no. 11, pp. 3787–3794, 2010.

[16] I. Karlsson, P. Papapetrou, and H. Boström, "Generalized random shapelet forests," *Data mining and knowledge discovery*, vol. 30, no. 5, pp. 1053–1085, 2016.

[17] W. Pei, H. Dibeklioğlu, D. M. Tax, and L. van der Maaten, "Multivariate time-series classification using the hidden-unit logistic model," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 4, pp. 920–931, 2018.

[18] M. G. Baydogan, "Time series representation and similarity based on local autopatterns," *Data Mining and Knowledge Discovery*, vol. 30, no. 2, pp. 476–509, 2016.

[19] P. Schäfer and U. Leser, "Multivariate time series classification with weasel+ muse," *arXiv preprint arXiv:1711.11343*, 2017.

[20] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *International Conference on Web-Age Information Management*. Springer, 2014, pp. 298–310.

[21] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate lstm-fcns for time series classification," *arXiv preprint arXiv:1801.04503*, 2018.

[22] A. Saltelli, "Sensitivity analysis for importance assessment," *Risk analysis*, vol. 22, no. 3, pp. 579–590, 2002.

[23] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," *arXiv preprint arXiv:1703.07015*, 2017.

[24] T. Sakai and M. Sugiyama, "Computationally efficient estimation of squared-loss mutual information with multiplicative kernel models," *IEICE TRANSACTIONS on Information and Systems*, vol. 97, no. 4, pp. 968–971, 2014.

[25] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015.

[26] "Offshore simulator centre," http://www.offsim.no/.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.