Journal of
Marine Science
and Engineering

MDPI

Article

Real-Time Prediction of Large-Scale Ship Model Vertical Acceleration Based on Recurrent Neural Network

Yumin Su, Jianfeng Lin, Dagang Zhao *, Chunyu Guo, Chao Wang and Hang Guo

College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China; suyumin@hrbeu.edu.cn (Y.S.); linjianfeng@hrbeu.edu.cn (J.L.); guochunyu_heu@outlook.com (C.G.); wangchao0104@hrbeu.edu.cn (C.W.); guohang1995@hrbeu.edu.cn (H.G.)

* Correspondence: zhaodagang@hrbeu.edu.cn

Received: 8 September 2020; Accepted: 1 October 2020; Published: 4 October 2020

check for updates

Abstract: In marine environments, ships are bound to be disturbed by several external factors, which can cause stochastic fluctuations and strong nonlinearity in the ship motion. Predicting ship motion is pivotal to ensuring ship safety and providing early warning of risks. This report proposes a real-time ship vertical acceleration prediction algorithm based on the long short-term memory (LSTM) and gated recurrent units (GRU) models of a recurrent neural network. The vertical acceleration time history data at the bow, middle, and stern of a large-scale ship model were obtained by performing a self-propulsion test at sea, and the original data were pre-processed by resampling and normalisation via Python. The prediction results revealed that the proposed algorithm could accurately predict the acceleration time history data of the large-scale ship model, and the root mean square error between the predicted and real values was no greater than 0.1. The optimised multivariate time series prediction program could reduce the calculation time by approximately 55% compared to that of a univariate time series prediction program, and the run time of the GRU model was better than that of the LSTM model.

Keywords: artificial intelligence; recursive neural network; time series prediction; large-scale ship model; vertical acceleration

1. Introduction

With the rapid economic development occurring worldwide, the scale of maritime transport is constantly expanding, and ship safety requirements are increasing. When a ship is at sea, its motion is affected by nonlinear winds, waves, currents, and other marine environment characteristics. In addition, the ship hull may be damaged under rough sea conditions. Therefore, obtaining real-time and accurate predictions of ship motion is vital to ensure ship safety and provide early warning of risks.

Over the last century, there has been steady advancement in the power of computers; simultaneously, data science and artificial intelligence have undergone rapid developments [1,2]. The effective combination of machine learning and artificial neural network (ANN) has been the focus of ship motion prediction research. The long short-term memory (LSTM) model is a recurrent neural network (RNN) used in machine learning, which has great potential in time series data prediction [3,4]. Karim et al. [5] applied the LSTM model to several complex multivariate time series classification tasks, such as activity recognition and action recognition. Srivastava et al. [6] studied the LSTM model as a powerful time series prediction method and investigated its potential in predicting solar irradiance. It was observed that the LSTM model outperformed the stringent machine learning benchmarks, proving its high accuracy in predicting irradiance. Wang et al. [7] simultaneously estimated the optical signal-to-noise ratio and nonlinear noise power caused by fibre nonlinearity through the LSTM model.

The aforementioned research results highlight the favourable prediction effect of the LSTM model on time series data and its high prediction speed. These factors satisfy the accuracy and real-time performance requirements in practice and can be used to predict ship performance.

The LSTM model has been widely used for predicting ship data. Gao et al. [8] used the LSTM model to predict ship tracks, proving that the LSTM model has the advantages of high accuracy, fast prediction, and easy realisation. Zhong et al. [9] used a two-way LSTM model to restore the missing tracks of inland river ships. Sun et al. [10] used an LSTM model to predict the position, speed, course, and other navigation parameters of a ship, verifying that the LSTM model could effectively generate navigation time series data for an automatic identification system. Moreover, the LSTM model provided a reliable basis for early warnings of ship collisions, search and rescue operations, and safety monitoring. Hu et al. [11] applied an LSTM model to the shipboard winch speed predictive control method, which addressed the poor control effect of an early neural network under complex working conditions.

In addition to the LSTM model, remarkable progress in ship-related predictions has been made using various neural network models. Yin et al. [12] designed an on-line prediction model of ship rolling motion based on a variable structure basis function neural network. Beşikçi et al. [13] developed a ship fuel consumption prediction system based on an ANN, which considers parameters such as ship speed, draught, propeller revolutions, and marine environment effects. Wang et al. [14] established an intelligent collision avoidance model for unmanned ships through a deep reinforcement learning obstacle avoidance decision-making algorithm based on the Markov decision process. Ferrandis et al. [15] compared the performance of a standard RNN with the gated recurrent units (GRU) and LSTM models by inputting random wave elevation under certain sea conditions and outputting the main motion of the ship, such as the pitch, heave, and roll. Gao et al. [16] employed the genetic algorithm to optimize the radial-based neural network to predict the load demand under fast-changing conditions. In addition, the authors used the Markov chain model to predict the load demand under slowly changing conditions to obtain the future load demand of the ship. Nagalingam et al. [17] presented an ensemble of extreme learning machine to estimate the longitudinal and side force coefficients as well as the yaw moment coefficient. Abebe et al. [18] used various machine learning regression techniques, such as the linear regression, polynomial regression, decision tree regressor, gradient boosting regressor, extreme gradient boosting regressor, random forest regressor, and extra trees regressor techniques, to predict ship speed. The models were finally used for actual ship route optimization purposes. Besides, the neural networks can be used for the implementation of continuous propeller torque demand prediction under rough sea conditions [19–21].

The existing studies on the real-time prediction of ship motion are primarily focused on full-scale ship rolling and track predictions; however, there is inadequate real-time prediction of large-scale ship model acceleration motion. In comparison to the time history data of ship rolling motion, the vertical acceleration time history curve of a ship model has a higher oscillation frequency and more complex nonlinear characteristics; thus, the time series prediction is more challenging. In this study, starting with the method of collecting vertical acceleration time history data for a large-scale ship model, a data pre-processing method based on Python was introduced. In addition, LSTM and GRU neural network models under the framework of TensorFlow2 were established and were respectively applied to the training and prediction of univariate and multivariate time series of the nonlinear vertical acceleration time history data of large-scale ship models at sea. Finally, the prediction results were compared and analysed.

2. Data Acquisition

In this study, the real-time prediction of the ship model vertical acceleration was performed based on the self-propulsion test data obtained through a large-scale ship model comprehensive test system at sea [22]. The experiment was conducted in the sea area near Qingdao, China (35.9° N, 120.2° E). As shown in Figure 1, the glass fibre-reinforced plastic large-scale ship model was a single machine, single propeller, bulbous bow, square stern ore carrier type. The main parameters for this ship model are summarised in Table 1.

Figure 1. Self-propulsion test of a large-scale ship model at sea.

Table 1. Main parameters of the large-scale ship model.

***Loa*, m**	***Lpp*, m**	***B*, m**	***D*, m**	***d*, m**	**∇, m^3**
24.99	24.20	4.04	1.87	1.39	115.2

Loa—overall length; *Lpp*—length between perpendiculars; *B*—breadth; *D*—depth; *d*—draft; ∇—displacement.

The overall layout of the related equipment on the large-scale ship model is shown in Figure 2. As demonstrated, the data acquisition system and control system were located in the cab; the propulsion system and auto-pilot system were installed in the aft cabin, and three vertical acceleration sensors were arranged in the bow, midship, and stern of the ship model (10% length between perpendiculars (*Lpp*), 50% *Lpp*, and 90% *Lpp*, respectively). The power of the self-propulsion ship model was provided by the propulsion system, including the generator, integrated cabinet, motor, and propeller. Through the console, Global Position System (GPS) and Inertial Navigation System (INS) antenna, and auto-pilot system, the ship speed and heading could be controlled. The self-propulsion instrument could obtain the thrust and torque of the propeller. In addition, the gyroscope could measure the roll and pitch angles of the ship model. The acceleration time history data (a_1, a_2, and a_3) obtained by the sensors were stored in a computer through the dynamic signal data collector, which provided the original data for the real-time prediction of the ship model vertical acceleration. The sensitivity of the accelerometer was 0.40 mA/EU, and the accuracy of the accelerometer was 10^{-6} m/s^2. The acquisition frequency of the data collector was 100 Hz. The frequency resolution corresponding to the fast Fourier transform (FFT) parameters was 0.0488 Hz. The FFT block length of the linear average spectrum analysis was 2000. The number of spectral lines was 800; the overlap rate was 0, and the acquisition mode was continuous recording. The inputs (a_1, a_2, and a_3) were obtained under the same conditions, as shown in Table 2. For a detailed introduction of the self-propulsion test of a large-scale ship model at sea, please refer to reference [22].

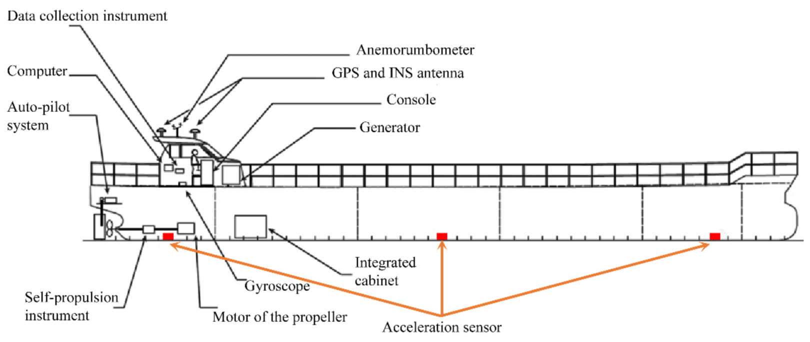

Figure 2. General arrangement of the relevant equipment on the large-scale ship model.

Table 2. Experimental environment and conditions.

Ship Speed, m/s	Wind Speed, m/s	Current Speed, m/s	Wave Period, s	Significant Wave Height, m	Sea Surface Temperature, °C
2.0	0.8	0.3	4.0	0.1	21

3. Data Pre-Processing

3.1. Resampling

Resampling is a process of converting the frequency of a time series, in which aggregating high-frequency data into low-frequency data is called downsampling and the inverse process is called upsampling [23]. The original data that were obtained from the large-scale ship model self-propulsion test in the real sea area contained 180,000 points. The training and prediction efficiency of the data would have been significantly affected by the excessive number of samples in the time series. Therefore, it was necessary to downsample and process the equal interval time calendar data of the vertical acceleration of the large-scale ship model. Based on the resampling method, the acquisition frequency of the acceleration time series was reduced from 100 Hz to 1 Hz. In addition, the mean value of the resampled packet data set was processed to use all the original time-history data.

3.2. Normalisation

Through the normalisation, different variables with different dimensions can be compared in terms of their numerical values, significantly improving the calculation accuracy. The objective of normalised scaling is to 'flatten' the data to a specified range, and the scaling size only involves the maximum and minimum values. In this study, the data collected by three vertical accelerometers were scaled uniformly to the range of (0, 1) and were converted into dimensionless data to improve the prediction accuracy of the neural network model. When outputting the variables, it was also necessary to perform anti-normalisation to restore the prediction sequence to the dimensional data.

4. Neural Network Model

It is suitable to use RNN for supervised learning problems with sequential data sets, such as time series forecasting [24]. As a generalisation of the feedforward neural network, a ring structure is added to the RNN to ensure that the output data of some neurons are fed back as input variables. The result of the RNN is a combination of the previous time input and historical input, which can be used to predict the time series; however, it will produce gradient divergence on the time axis. The standard RNN model has both exploding and vanishing gradients, which are caused by the iterative property of an RNN, whose gradient is essentially equal to the higher power of the recursive weight matrix. The power of these iterative matrices leads to exponential growth or contraction of the gradient in time steps. Simply reducing the gradients whose norms exceed a certain threshold makes the explosive gradient problem relatively easy to handle; this technique is called gradient clipping. If the gradient is frequently reduced by a large factor, learning will be affected, but if the norm of the gradient is small most of the time, gradient clipping is very effective. The vanishing gradient issue is more challenging because it does not cause the gradient itself to become smaller. Although the component of the gradient is small in the direction corresponding to the long-term correlation, it is large in the direction corresponding to the short-term correlation. Consequently, the RNN model can learn short-term dependences more easily than long-term dependences [25].

Two popular and efficient improved RNN models showing excellent performance are the LSTM and GRU models, which can learn and remember the characteristics of time series and avoid gradient divergence. In this study, the LSTM and GRU models were used as the basic neural networks of the real-time prediction algorithm of the large-scale ship model vertical acceleration to address the aforementioned problems. Figure 3 shows a flow chart explaining the process from data collection to the pre-processing and post-processing results.

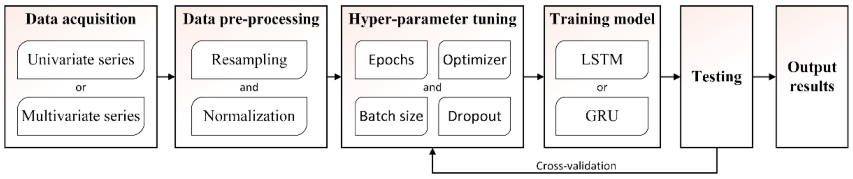

Figure 3. Scheme of the methodology.

4.1. LSTM Model

In the LSTM model, the neural network state is divided into long-term and short-term types. The long-term state c_t is unique to the LSTM model, whereas the short-term state h_t resembles an ordinary RNN. Moreover, h_t is the output of the current hidden layer and input of the next hidden layer. The architecture of the LSTM model in a single time step is shown in Figure 4, which includes three gating units, namely, the input gate i_t, forget gate f_t, and output gate o_t. Among these gating units, i_t and f_t are fundamental to determining whether the LSTM model can achieve long-term and short-term memory. The LSTM model unit has three inputs, namely, the current time input x_t, previous long-term state c_{t-1}, and short-term state h_{t-1}, where x_t and h_{t-1} are simultaneously input into the three gating units. f_t determines the number of previous memory values that should be removed from the cell state, whereas i_t specifies the new input to h_{t-1}. In addition, o_t determines the data output to h_t at the current time [26].

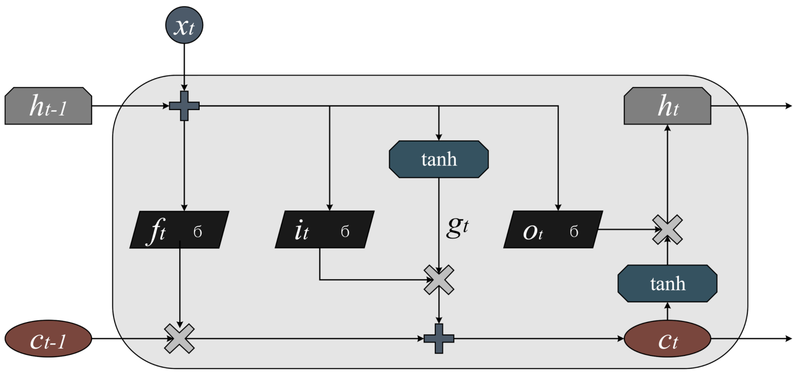

Figure 4. Long short-term memory (LSTM) model structure in a single time step (where c_t is the long-term state, h_t is the short-term state, i_t is the input gate, f_t is the forget gate, o_t is the output gate, x_t is the current time input, c_{t-1} is the previous long-term state, h_{t-1} is the previous short-term state, σ is the sigmoid function, and tanh is the activation function of the hidden layer).

The LSTM model can realise the memory function of the vertical acceleration time series of a large-scale ship model through the switch of the gate control unit and can prevent gradient divergence. Therefore, it can be used to deal with and perform predictions in problems that have long time intervals and delays. Concurrently, the LSTM model can deal with noise, distributed representation, and continuous values [27]. The specific calculation formula of the LSTM model is as follows [28]:

$$i_t = \sigma\left(W_{hi}^T \cdot h_{t-1} + W_{xi}^T \cdot x_t + b_i\right), \tag{1}$$

$$f_t = \sigma\left(W_{hf}^T \cdot h_{t-1} + W_{xf}^T \cdot x_t + b_f\right), \tag{2}$$

$$o_t = \sigma\left(W_{ho}^T \cdot h_{t-1} + W_{xo}^T \cdot x_t + b_o\right), \tag{3}$$

$$g_t = \tanh\left(W_{hg}^T \cdot h_{t-1} + W_{xg}^T \cdot x_t + b_g\right), \tag{4}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes g_t, \tag{5}$$

$$h_t = o_t \otimes \tanh(c_t), \tag{6}$$

where σ is the sigmoid function; tanh is the activation function of the hidden layer; g_t is the output of the middle layer; W_{hi}, W_{hf}, W_{ho}, and W_{hg} are the weight matrices that are connected to h_{t-1} for each layer; W_{xi}, W_{xf}, W_{xo}, and W_{xg} are the weight matrices that are connected to x_t for each layer; and b_i, b_f, b_o, and b_g are the bias vectors of the layers.

4.2. GRU Model

Compared with the LSTM model proposed in 1997, the network structure of the GRU model designed in 2014 is simpler, with only two gated units (an update gate z_t and a reset gate r_t) and without a storage unit [29]. Specifically, i_t and f_t in the LSTM model are merged into z_t without o_t. This combination is achieved by introducing a linear dependency between the current network state H_t and the previous network state H_{t-1}, without distinguishing between the long and short states, to solve the gradient divergence problem of the RNN. The architecture of the GRU model in a single time step is presented in Figure 5. z_t controls the amount of H_{t-1} to be saved in H_t and the amount of the candidate state G_t to retain in the current time. Meanwhile, r_t combines the current time input x_t with H_{t-1}, and it specifies the degree of inheritance of G_t to H_{t-1}.

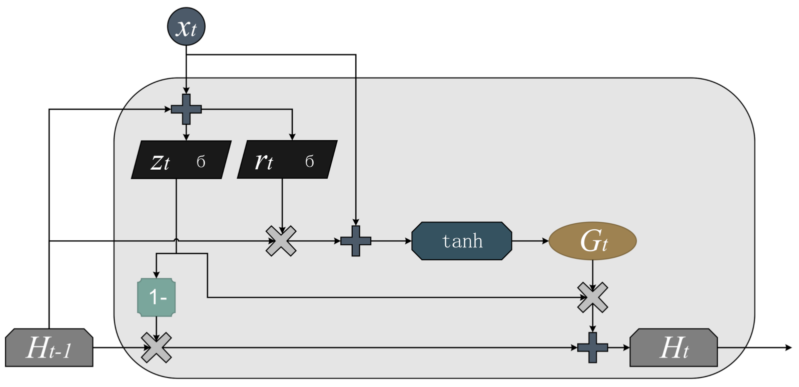

Figure 5. Gated recurrent units (GRU) model structure in a single time step (where z_t is the update gate, r_t is the reset gate, x_t is the current time input, H_t is the current network state, H_{t-1} is the previous network state, G_t is the candidate state, σ is the sigmoid function, and tanh is the activation function of the hidden layer).

The GRU model can be regarded as a variant of the LSTM model, which simplifies the neural network structure and retains the effect of the LSTM model. It is suitable for situations with low hardware conditions or high time cost. The specific calculation formulas for the GRU model are as follows [28]:

$$z_t = \sigma\left(W_{Hz}^T \cdot H_{t-1} + W_{xz}^T \cdot x_t\right), \tag{7}$$

$$r_t = \sigma\left(W_{Hr}^T \cdot H_{t-1} + W_{xr}^T \cdot x_t\right), \tag{8}$$

$$G_t = \tanh\left(W_{HG}^T \cdot (r_t \otimes H_{t-1}) + W_{xG}^T \cdot x_t\right), \tag{9}$$

$$H_t = (1 - z_t) \otimes \tanh\left(W_{xG}^T \cdot H_{t-1} + z_t \otimes G_t\right), \tag{10}$$

where W_{Hz}, W_{Hr}, and W_{HG} are the weight matrices that are connected to H_{t-1} for each layer and W_{xz}, W_{xr}, and W_{xG} are the weight matrices that are connected to x_t for each layer.

5. Results and Discussion

In this experiment, the common open-source third-party libraries Pandas and NumPy that use Python language and the TensorFlow2 deep learning framework developed by Google were utilised.

The program ran on a desktop computer graphics processing unit (GPU). The specific configuration of the computer is presented in Table 3. Based on the LSTM and GRU models, the univariate and optimised multivariate time series prediction models were validated and compared, and the prediction error performance index of each model was studied.

Table 3. Main parameters of the computer.

Operating System	Windows 7 64 Bits
Central processing unit (CPU) model	Intel Core i5-6500
Graphics processing unit (GPU) model	NVIDIA GeForce GT 720
CPU frequency, GHz	3.2
Memory size, GB	2.0
NVIDIA GPU compute capability	3.5

Three groups of pre-processed vertical acceleration time series of the large-scale ship model were used as the input datasets of the LSTM and GRU models. The first 60% of the time series data in each set were used for training, and the remaining 40% of the data were used for prediction. After testing and optimising the hyper-parameters of the neural network model, the numbers of hidden layer nodes and training epochs in this experiment were 50 and 500, respectively, and the batch size was 128. By using the Adam optimiser, the objective function to be minimised in the training process was the mean square error. A regularisation method commonly used in deep learning, called dropout, was used to deal with the prediction data to avoid the over-fitting phenomenon. In the process of neural network training, some of the hidden layer neurons were randomly removed; the fully connected network was sparse, and the complex co-adaptability between the neurons was reduced. Only the remaining neurons were used to build a new neural network, enhancing the robustness of the algorithm model [30]. However, the dropout method has a disadvantage in the sense that it will lead to decreased peak prediction accuracy. Owing to the pitching motion of the ship model at sea, the peak accelerations of the bow and stern of the ship model are certainly larger than those of the midship under the same working conditions, resulting in a large overshoot.

Table 4 lists the input and output variables corresponding to 12 sets of forecast conditions. In addition, the table presents the total run times, root mean square errors (RMSEs), and coefficients of determination (R^2) of the different programs. For the univariate time series prediction program, the RMSE of the GRU model is lower than that of the LSTM model, the total run time of the GRU model is reduced by approximately 8%, and the R^2 of the GRU model is increased by approximately 40%. These results are related to the simpler network structure of the GRU model mentioned in Section 4.2. For the multivariate time series prediction program, the RMSE of the GRU model is slightly higher than that of the LSTM model; however, its time consumption is low and R^2 is higher. Moreover, the time consumption of the multivariate time series prediction program can be reduced by approximately 55% compared to that of the univariate time series prediction program, its RMSE can be reduced by approximately 20%, and its R^2 can be increased by approximately 2.5 times. Thus, this approach has advantages in terms of time cost and calculation accuracy. Furthermore, the time cost can be further reduced by enhancing the hardware performance, such as the GPU computational capability.

Comparisons between the predicted and actual vertical acceleration time history data of the large-scale ship model are shown in Figures 6–8. Combined with Table 4, the RMSEs of the LSTM and GRU models are observed to be approximately equal and lower than 0.1, which verifies the reliability of the neural network model and the algorithm implemented in this study. The overall trend of the acceleration data can be predicted accurately in each case, and the multivariate time series prediction algorithm can maintain a favourable prediction effect when the acceleration amplitude changes significantly. Because the predicted values of the acceleration time series of the bow, midship, and stern cannot accurately capture the actual high amplitude, the RMSE of its prediction program is higher than that obtained from the ship data. Although there are some differences in both the R^2 values and the plots presented, the error mainly originates from the prediction of the peak value, and the prediction RMSE of the algorithm remains acceptable. Figure 9 considers case 8 as an example to show the loss curve of the neural network model training and prediction. The final loss is approximately 0.02. Moreover, no over-fitting occurs, which further verifies the accuracy of the prediction results.

Table 4. Forecast conditions.

Cases	Neural Network Model	Input Variable	Output Variable	Total Run Time, s	RMSE	R^2
1	LSTM	a_1	a_1	76.71	0.093	0.189
2	GRU	a_1	a_1	70.54	0.091	0.293
3	LSTM	a_2	a_2	76.67	0.061	0.116
4	GRU	a_2	a_2	70.62	0.060	0.154
5	LSTM	a_3	a_3	76.75	0.087	0.179
6	GRU	a_3	a_3	70.76	0.085	0.231
7	LSTM	a_1, a_2, a_3	a_1	34.01	0.077	0.510
8	GRU	a_1, a_2, a_3	a_1	32.95	0.079	0.535
9	LSTM	a_1, a_2, a_3	a_2	33.58	0.043	0.599
10	GRU	a_1, a_2, a_3	a_2	32.65	0.043	0.601
11	LSTM	a_1, a_2, a_3	a_3	33.47	0.069	0.610
12	GRU	a_1, a_2, a_3	a_3	32.92	0.071	0.617

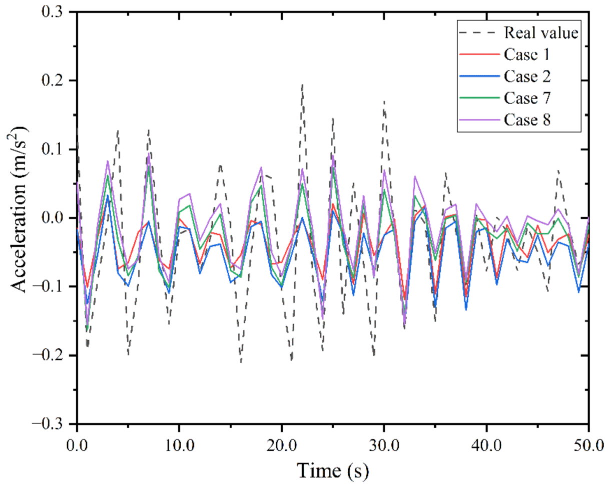

Figure 6. Predicted bow acceleration.

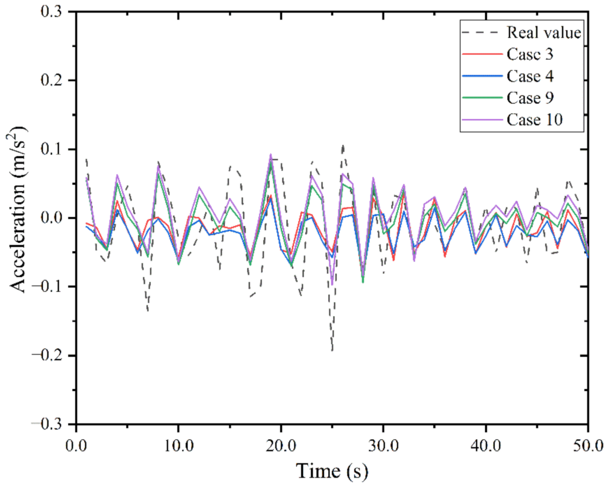

Figure 7. Predicted midship acceleration.

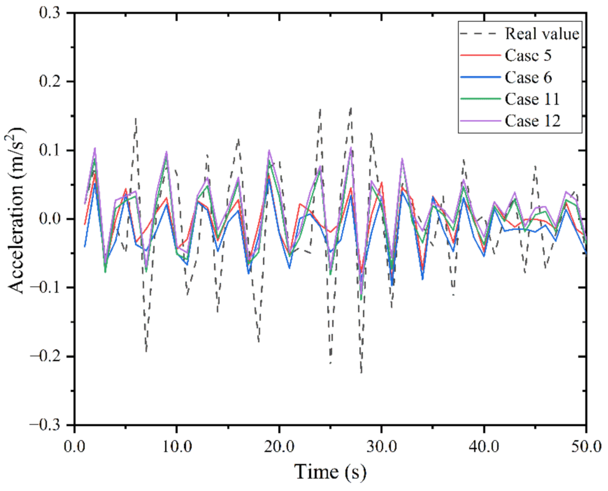

Figure 8. Predicted stern acceleration.

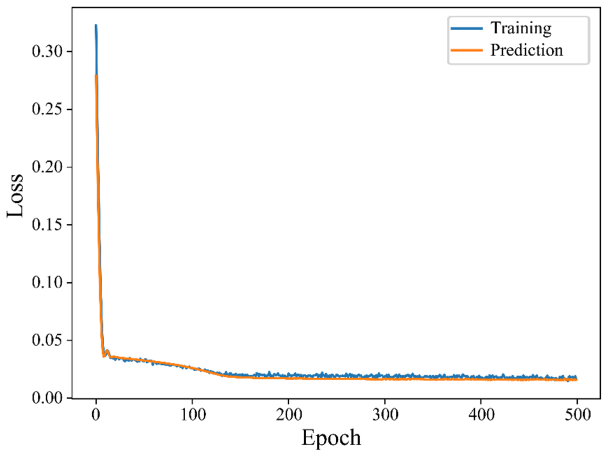

Figure 9. Loss comparison (Case 8).

6. Conclusions

Accurate prediction of the ship motion state is essential to ensure the safety of ships and provide early warning of risks. This report proposed a real-time ship model vertical acceleration prediction algorithm based on the LSTM and GRU models. Through pre-processing operations, such as resampling, and normalisation of the vertical acceleration time series of a large-scale ship model, the data structure could be simplified and the computational efficiency of the algorithm could be significantly improved. By comparing and analysing the prediction results of the different neural network models with univariate and multivariate inputs, the following conclusions can be drawn:

(1) The proposed algorithm can accurately predict the acceleration time history data of large-scale ship models at sea. In addition, the RMSE between the predicted and actual values was less than 0.1, the local prediction accuracy was affected by the high amplitude of the time history data, the final loss for neural network model training and prediction was approximately 0.02, and no over-fitting occurred.

(2) The optimised multivariate time series prediction program could reduce the computing time by approximately 55% compared to that of the single-variable time series prediction program. In addition, the GRU model presented several advantages in terms of simplifying the neural network and improving the run time.

Consequently, multivariate time series prediction algorithm based on the GRU model has better application value in the actual environment that was considered in this study. The algorithm proposed in this report can predict future acceleration values even tens of seconds in advance. Combined with the risk assessment procedure, safety warning can be provided, and measures, such as reducing the speed, changing the course, and starting an anti-rolling device can be considered in advance. The acceleration can be utilized to evaluate the ship strength and seakeeping performance by analysing the hydroelastic responses and obtaining the response amplitude operators using Fourier analysis. Additionally, the algorithm can be employed to make predictions for other artificial intelligence systems, including motion prediction, risk assessment, and structural deformation detection. Future studies need to focus on improving the generalisation ability and stability of the LSTM and GRU models. Finally, more key parameters should be introduced into the prediction model in future research, and these parameters should be combined with more variables to obtain more accurate prediction results.

Author Contributions: Conceptualization, Y.S. and C.G.; software, J.L.; validation, D.Z., J.L., and C.W.; investigation, J.L.; resources, Y.S.; data curation, D.Z.; writing—original draft preparation, J.L.; writing—review and editing, D.Z.; visualization, J.L. and H.G.; supervision, Y.S. and C.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number 51709060).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jurdana, I.; Krylov, A.; Yamnenko, J. Use of artificial intelligence as a problem solution for maritime transport. *J. Mar. Sci. Eng.* **2020**, *8*, 201. [CrossRef]
2. Li, X.; Yan, Z.; Liu, Z. Combination and application of machine learning and computational mechanics. *Chin. Sci. Bull.* **2019**, *64*, 635–648. [CrossRef]
3. Marchi, E.; Vesperini, F.; Eyben, F.; Squartini, S.; Schuller, B. A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks. In Proceedings of the ICASSP IEEE International Conference on Acoustics, Speech, and Signal Processing, Brisbane, QLD, Australia, 19–24 April 2015. [CrossRef]
4. Shi, Z.; Xu, M.; Pan, Q.; Yan, B.; Zhang, H. LSTM-based flight trajectory prediction. In Proceedings of the IEEE 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8. [CrossRef]
5. Karim, F.; Majumdar, S.; Darabi, H.; Harford, S. Multivariate LSTM-FCNs for time series classification. *Neural Netw.* **2019**, *116*, 237–245. [CrossRef] [PubMed]
6. Srivastava, S.; Lessmann, S. A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data. *Sol. Energy* **2018**, *162*, 232–247. [CrossRef]
7. Wang, Z.; Yang, A.; Guo, P.; He, P. OSNR and nonlinear noise power estimation for optical fiber communication systems using LSTM based deep learning technique. *Opt. Express* **2018**, *26*, 21346–21357. [CrossRef] [PubMed]
8. Gao, M.; Shi, G.; Li, S. Online prediction of ship behavior with automatic identification system sensor data using bidirectional long short-term memory recurrent neural network. *Sensors* **2018**, *18*, 4211. [CrossRef] [PubMed]
9. Zhong, C.; Jiang, Z.; Chu, X.; Liu, L. Inland ship trajectory restoration by recurrent neural network. *J. Navig.* **2019**, *72*, 1359–1377. [CrossRef]
10. Yang, S.; Xinya, P.; Zexuan, D.; Jiansen, Z. An approach to ship behavior prediction based on AIS and RNN optimization model. *Int. J. Transp. Eng. Technol.* **2020**, *6*, 16–21. [CrossRef]
11. Hu, J.; Guo, J.; Ding, B. Automatic control of paving ship based on LSTM. In Proceedings of the 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Jinzhou, China, 6–8 June 2019. [CrossRef]
12. Yin, J.C.; Zou, Z.J.; Feng, X. On-line prediction of ship roll motion during maneuvering using sequential learning RBF neural networks. *Ocean Eng.* **2013**, *61*, 139–147. [CrossRef]
13. Besikci, E.B.; Arslan, O.; Turan, O.; Olcer, A.I. An artificial neural network based decision support system for energy efficient ship operations. *Comput. Oper. Res.* **2016**, *66*, 393–401. [CrossRef]
14. Wang, C.; Zhang, X.; Cong, L.; Li, J.; Zhang, J. Research on intelligent collision avoidance decision-making of unmanned ship in unknown environments. *Evol. Syst.* **2018**, *10*, 649–658. [CrossRef]

15. del Águila Ferrandis, J.; Triantafyllou, M.; Chryssostomidis, C.; Karniadakis, G. Learning functionals via LSTM neural networks for predicting vessel dynamics in extreme sea states. *arXiv* **2019**, arXiv:1912.13382.
16. Gao, D.; Jiang, Y.; Zhao, N. A novel load prediction method for hybrid electric ship based on working condition classification. *Trans. Inst. Meas. Control* **2020**. [CrossRef]
17. Nagalingam, K.K.; Savitha, R.; Mamun, A.A. An ensemble of Extreme Learning Machine for prediction of wind force and moment coefficients in marine vessels. In Proceedings of the 2016 IEEE International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 2901–2908. [CrossRef]
18. Abebe, M.; Shin, Y.; Noh, Y.; Lee, S.; Lee, I. Machine learning approaches for ship speed prediction towards energy efficient shipping. *Appl. Sci.* **2020**, *10*, 2325. [CrossRef]
19. Xiros, N.I.; Kyrtatos, N.P. A neural predictor of propeller load demand for improved control of diesel ship propulsion. In Proceedings of the 2000 IEEE International Symposium on Intelligent Control. Held Jointly with the 8th IEEE Mediterranean Conference on Control and Automation (Cat. No.00CH37147), Rio Patras, Greece, 17–19 July 2000; pp. 321–326. [CrossRef]
20. Kyrtatos, N.P.; Theotokatos, G.; Xiros, N.I.; Marek, K.; Duge, R. Transient Operation of Large-bore Two-stroke Marine Diesel Engine Powerplants: Measurements & Simulations. In Proceedings of the 23rd CIMAC Congress, Hamburg, Germany, 7–10 May 2001.
21. Kytatos, N.P.; Theotokatos, G.; Xiros, N.I. Main Engine Control for Heavy Weather Conditions. In Proceedings of the ISME 6th International Symposium on Marine Engineering, Tokyo, Japan, 23–27 October 2000.
22. Lin, J.; Zhao, D.; Guo, C.; Su, Y.; Zhong, X. Comprehensive test system for ship-model resistance and propulsion performance in actual seas. *Ocean Eng.* **2020**, *197*, 106915. [CrossRef]
23. Hu, R.; Chiu, Y.-C.; Hsieh, C.-W.; Chang, T.-H.; Xue, X.; Zou, F.; Liao, L. Mass rapid transit system passenger traffic forecast using a re-sample recurrent neural network. *J. Adv. Transport.* **2019**, *2019*, 8943291. [CrossRef]
24. Petneházi, G. Recurrent neural networks for time series forecasting. *arXiv* **2019**, arXiv:1901.00069v1.
25. Jozefowicz, R.; Zaremba, W.; Sutskever, I. An empirical exploration of recurrent network architectures. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 2342–2350.
26. Qing, X.; Niu, Y. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy* **2018**, *148*, 461–468. [CrossRef]
27. Sepp, H.; Jürgen, S. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
28. Géron, A. *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly: Beijing, China, 2017.
29. Irie, K.; Tuske, Z.; Alkhouli, T.; Schluter, R.; Ney, H. *LSTM, GRU, Highway and a Bit of Attention: An Empirical Overview for Language Modeling in Speech Recognition*; Interspeech: San Francisco, CA, USA, 2016; pp. 3519–3523. [CrossRef]
30. Billa, J. Dropout approaches for LSTM based speech recognition systems. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5879–5883. [CrossRef]