



Article

Roll Motion Prediction of Unmanned Surface Vehicle Based on Coupled CNN and LSTM

Wenjie Zhang ¹ , Pin Wu ^{1,*}, Yan Peng ² and Dongke Liu ²

¹ School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China; wendy_zhang@shu.edu.cn

² School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China; pengyan@shu.edu.cn (Y.P.); liudongke@shu.edu.cn (D.L.)

* Correspondence: wupin@shu.edu.cn; Tel.: +86-1500-071-6875

Received: 14 October 2019; Accepted: 3 November 2019; Published: 18 November 2019



Abstract: The prediction of roll motion in unmanned surface vehicles (USVs) is vital for marine safety and the efficiency of USV operations. However, the USV roll motion at sea is a complex time-varying nonlinear and non-stationary dynamic system, which varies with time-varying environmental disturbances as well as various sailing conditions. The conventional methods have the disadvantages of low accuracy, poor robustness, and insufficient practical application ability. The rise of deep learning provides new opportunities for USV motion modeling and prediction. In this paper, a data-driven neural network model is constructed by combining a convolution neural network (CNN) with long short-term memory (LSTM) for USV roll motion prediction. The CNN is used to extract spatially relevant and local time series features of the USV sensor data. The LSTM layer is exploited to reflect the long-term movement process of the USV and predict roll motion for the next moment. The fully connected layer is utilized to decode the LSTM output and calculate the final prediction results. The effectiveness of the proposed model was proved using USV roll motion prediction experiments based on two case studies from “JingHai-VI” and “JingHai-III” USVS of Shanghai University. Experimental results on a real data set indicated that our proposed model obviously outperformed the state-of-the-art methods.

Keywords: CNN; data-driven; LSTM; roll motion prediction; unmanned surface vehicle

1. Introduction

Unmanned surface vehicles (USVs) [1] are small unmanned marine vehicles that travel on water in a remote or autonomous manner. USVs are characterized by small size, fast movement, and high flexibility. In recent years, they have been adopted to conduct missions such as marine rescue, environmental monitoring [2], island reef mapping [3], and resource exploration [4]. The safety of USVs is critical when conducting missions, and the roll motion is directly related to the safety and operating performance [5]. In order to ensure that USVs conduct their missions safely, it is necessary to predict the USV roll motion, so that the operator or automatic control system has sufficient time to avoid serious accidents.

The USV roll motion at sea is a time-varying, nonlinear, and uncertain complex dynamic system. It is affected by control systems such as steering and propulsion systems, as well as external disturbances produced by wind, waves, and sea current [6,7]. Moreover, the USV roll motion and the movements in other degrees of freedom (e.g., heave and pitch motions) are coupled with each other. Therefore, it is difficult to establish precise mathematical models to represent the USV roll motion at sea.

Over the past decades, many methods have been developed to model and predict ship motion. Francescutto et al. [8] applied an available mathematical model with concentrated parameters to

the roll-sloshing problem. Daalen et al. [9] formulated a differential equation model (a Mathieu-type equation) for modeling the roll dynamics of a ship sailing in large-amplitude head waves. Silva et al. [10] described a time-domain non-linear strip theory model of ship motions in six degrees of freedom based on a potential flow strip theory using Frank's close fit method. These methods require mechanical analysis and mathematical modeling of the hull, so there are some difficulties in practical application. Later, scholars proposed some simple and practical methods. According to the theoretical differences, these methods can be classified into three types: Kalman filtering, time series methods, and neural-network-based methods [11]. Kalman filtering is a recursive linear minimum variance filter for online forecasting [12]. Triantafyllou et al. [13] proposed a Kalman filtering model for ship motion prediction and applied it to a DD-963 destroyer. In implementing the Kalman filtering, accurate state-space equations and noise statistics are necessary. However, in practical engineering applications, these are difficult to obtain. Therefore, although Kalman filtering is simple in calculation, its use is difficult in practical applications.

Time series methods provide a feasible solution, which only requires the history and current ship motion status data for prediction. These methods are suitable for practical engineering applications, as they do not require a comprehensive understanding of the ship's dynamic system. Classic time series prediction methods which are widely used include autoregressive (AR) and moving average (MA) models, as well as many extended models based on them [14–16]. Yumori et al. [17] proposed a time domain model based on autoregressive moving average (ARMA) to predict real-time ship motion. This model best fitted an input wave sensor time history to the ship response time history and was applied to aircraft landing on the ship. However, as ship motion is non-stationary, it conflicts with the stationary assumption of classic time series methods. Therefore, many improved methods based on classic time series methods have been proposed. Zhou et al. [18] conducted a nonlinear autoregressive (NAR) model using an orthogonalization technique, and the experimental results indicated that the NAR model gave better prediction accuracy than the AR model. Jun et al. [19] combined empirical mode decomposition (EMD) and discrete wavelet transform decomposition (DWT) to improve the AR model for ship motion prediction. Compared with the conventional AR model, this model is more capable of handling nonlinear and non-stationary signals. Suhermi et al. [20] adapted a hybrid methodology to combine autoregressive integrated moving average (ARIMA) and deep neural network (DNN) models for predicting the roll motion. The hybrid model had a good ability to capture the linear and nonlinear patterns. Although these improved methods have shown their effectiveness, they are still limited in nonlinear ship motion prediction, as explicit relationships between input and output variables for the data sets have to be hypothesized.

Neural networks are a simulation of the biological nervous system [21]. In contrast to time series methods, neural networks are more capable of modeling nonlinear systems without a priori knowledge of the relationships between input and output variables. In recent decades, neural networks have been successfully applied in various fields, such as natural language processing [22], computer vision [23], and autonomous driving [24]. There are also some USV applications. Wang et al. [25] presented a path-following controller based on a radial basis function neural network (RBFNN) for formation control in single unmanned surface vehicles. In theory, a neural network can approximate any nonlinear system with arbitrary accuracy [26]. Accordingly, they have been used to model the nonlinear ship motion dynamic system for prediction. Yang et al. [27] obtained a ship motion prediction model based on a BP neural network by training the ship motion MATLAB simulation data. Their experimental results showed that the trends of the true and predicted values were consistent. Huang et al. [28] proposed a coarse and fine-tuning fixed-grid wavelet network model to predict ship roll motion, which is trained by simulated ship roll motion data in regular waves. These neural network models are trained by MATLAB simulation data instead of actual measured sensor data, and therefore the applicability of these models in the real world remains to be verified.

Yin et al. [29] constructed an ensemble prediction scheme by combining the discrete wavelet transform (DWT) method with a variable-structure radial basis function (RBF) network for real-time

ship roll motion prediction. Yin et al. [30] constructed a variable-structure RBF network sequentially by an adaptive sliding data window (SDW) learning scheme to predict ship roll motion online. The two references adopted a variable-structure RBF neural network as the prediction model. In their model, an innovative algorithm was proposed to adjust the data sliding window online. Both the structure and parameters of RBF neural network were tuned according to the data of the current window in order to achieve real-time prediction. However, in the process of training the neural network, they used real-time data, making the training set relatively small. In this way, the neural network model could be overfitting and its generalization ability was poor. Moreover, they only used ship roll motion time series data to train the neural network model, and ignored the coupling characteristics between the ship movements in six degrees of freedom [31,32], as well as the influence of the control system on the ship's movements.

With the rise of deep learning (DL) [33], data-driven prediction methods are increasingly applied in various fields [34,35]. In maritime applications, Joohyun Woo [36] proposed a deep-learning-based dynamic model identification method. The long short-term memory (LSTM) [37] based dynamic model extracted patterns of USV dynamics from free-running test data, and outperformed conventional simplified maneuvering models. However, the data-driven method requires a large amount of data for training. Fortunately, there are normally many sensors installed on the ship at various positions. During the sea trials, these sensors measure the ship's motion status data and control system status data in real time. These time series sensor data are stored. Therefore, this approach provides a new strategy for ship motion modeling and prediction. We can model the ship motion by mining the hidden information of sensor data.

In this paper, we propose a coupled convolutional neural network (CNN) [38] and long short-term memory (LSTM) [37] model for USV roll motion prediction. The data measured by sensors installed on the USV are adopted as the data set of the proposed model. These sensor data contain six-degree-of-freedom motion status data and control status data, and constitute a multidimensional space of the USV movement. The USV roll motion of the next moment is influenced not only by the other five degrees of freedom and control states, but also by the movement of the previous period. CNNs have been proven to be powerful for processing spatial data, and have been widely used in computer vision [38–40]. LSTM is a type of recurrent neural network (RNN) [41] designed for the time-series problem. Therefore, in our proposed model, a CNN is used to extract spatial features and local time-series features of the USV sensor data. The output of the CNN layer are higher-dimensional feature maps, which are the input of the LSTM layer. The wind, waves, and sea current are natural phenomena that usually change continuously over a period of time. They act on the USV, causing a series of changes in the USV's motion status. So, the current USV roll motion is affected by the motion status in the past. Therefore, the LSTM layer is exploited to model the long-term movement process of the USV and predict the roll motion in the next moment. After that, the fully connected layer is utilized to decode the LSTM output and calculate the final prediction results. The proposed model is able to extract features in both spatial and temporal dimensions to obtain better prediction results for USV roll motion. To prove the effectiveness of the proposed model, it was applied to "JingHai-VI" and "JingHai-III" USVs of Shanghai University.

The paper is organized as follows: Section 2 presents the proposed coupled CNN and LSTM model for USV roll motion prediction in detail. Section 3 describes the source of the data set and the data preprocessing process. The experimental results and discussion of two case studies are shown in Section 4. Finally, the paper is concluded in Section 5.

2. Methodology

2.1. Framework of the Proposed Prediction Model

In this paper, a coupled CNN and LSTM model is proposed for USV roll motion prediction. The target roll motion time series data are denoted as $R = \{r_{t-D}, r_{t-D+1}, \dots, r_t\}$. The

other degree-of-freedom movement status data and control status data are denoted as $F^i = \{f_{t-D}^i, f_{t-D+1}^i, \dots, f_t^i\}$, where t and D denote the time step and the length of setting time window, respectively. In our proposed prediction model, the input time series data is $S = R \cup F^1 \cup F^2 \cup \dots \cup F^i$, and the output is the target roll motion data at time step $t + 1$, denoted as r_{t+1} . The framework of the proposed model is shown in Figure 1. The multi-channel one-dimensional convolution layer is utilized to eliminate data redundancy and extract spatially relevant features. The LSTM layer is exploited to extract the long-term time-series features and model the USV roll motion. The fully connected layer is used to decode the LSTM output and calculate the final predicted results at the next moment. The input data are mapped into higher-dimensional feature maps, and these feature maps are the input to the LSTM layer.

Our proposed prediction model for USV roll motion can be expressed by the following:

$$C_t = f(W^c X_t), \tag{1}$$

$$L_t = g(W^l C_t + U^l L_{t-1}), \tag{2}$$

$$Y_t = h(W^y L_t), \tag{3}$$

where X_t denotes a set of input data and Y_t represents the output of the prediction model. C_t , L_t , and Y_t are the outputs of the convolution layer, LSTM layer, and fully connected layer, respectively. W^c , W^l , U^l , and W^y are the weight matrices of the model: W^c denotes the weight matrices of the input layer to the convolutional layer; W^l denotes the weight matrices of the convolutional layer to the LSTM layer; U^l is the weight matrices of the LSTM hidden layer at last moment to the output of the current moment; W^y means the weight matrices of the LSTM layer to the fully connected layer.

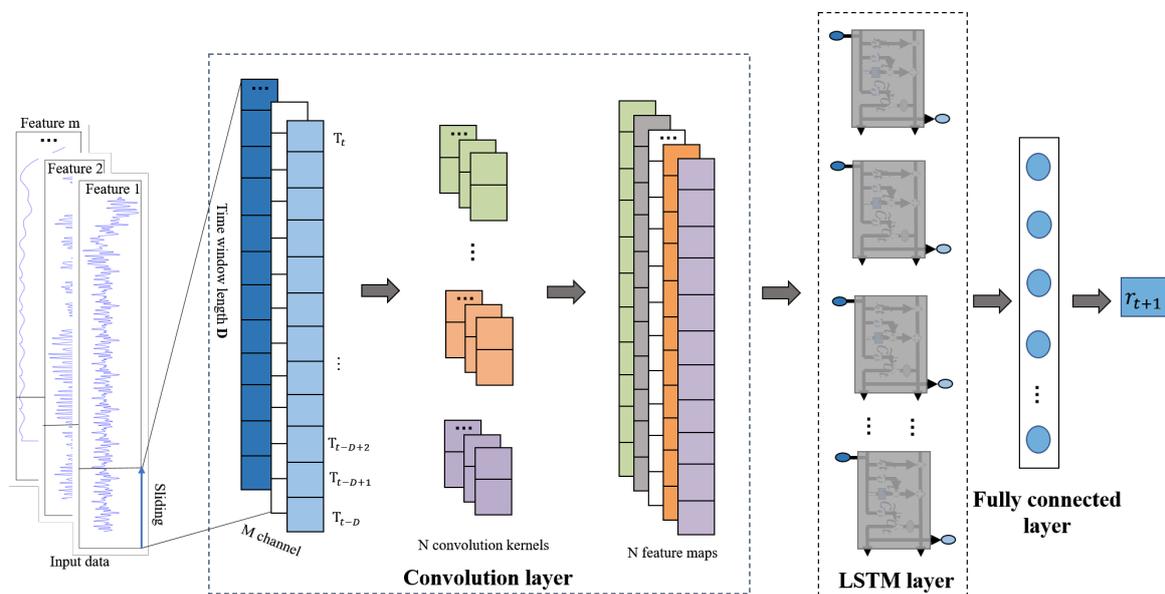


Figure 1. Framework of the proposed coupled convolutional neural network (CNN) and long short-term memory (LSTM) prediction model.

2.2. Convolution-Based USV Sensor Data Feature Extraction

A typical CNN is alternately performed by several convolution layers, pooling layers, and fully connected layers, as shown in Figure 2. The convolution layer extract features from the input data by performing a convolution operation. This is the core of the CNN. For one-dimensional input data, the convolution operation with a one-dimensional kernel is performed as follows:

$$s(t) = (x * \omega)(t) = \sum_a x(a)(t - a), \tag{4}$$

where x is the input data, t represents time, ω denotes weighting function (also known as the convolution kernel), a is the position where the convolution kernel is currently sliding, and $*$ denotes the convolution operation. Feature mapping is implemented by sliding several different convolution kernels. The parameters are shared with each convolution kernel on each convolution layer. The weight-sharing structure of the CNN reduces the network complexity and reduces the number of weights. The pooling layer reduces the size of the output dimension by max pooling or average pooling. After several rounds of convolution and pooling, the multidimensional data will be flattened by the fully connected layer. At present, the typical CNN has been improved according to different tasks.

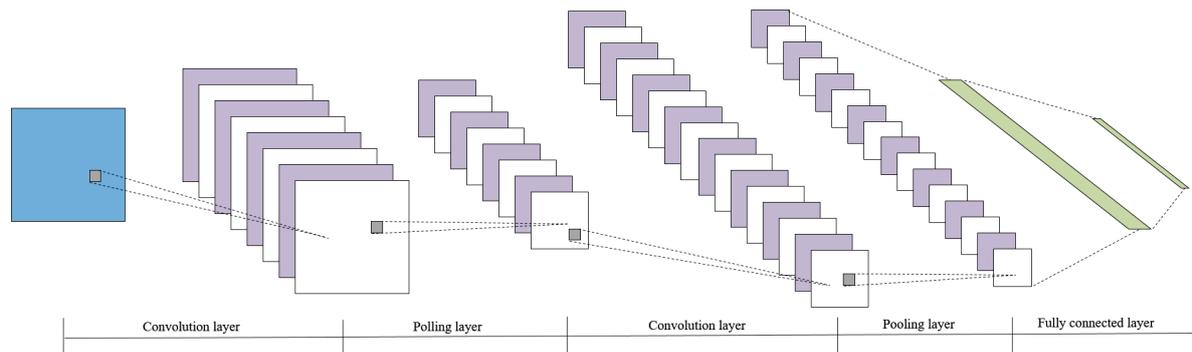


Figure 2. The typical CNN structure.

In our proposed prediction model, the input is the sensor data of the USV. These sensor data are different features of USV movements at a certain point in time. These features constitute a multidimensional space of the USV’s movements. Similar to the color images, which are a three-dimensional space composed of three channels (RGB), these sensor data can be regarded as a multi-dimensional space composed of multi-channel movement features. Inspired by the CNN feature extraction of color images, in the proposed model, we automatically extract features of these sensor data through the convolution operation.

At the CNN layer, the input data are converted to several three-dimensional matrices of $1 \times D \times m$ (also known as M -dimensional row vector). D denotes the length of the setting time window; m is the number of channels (also known as the number of features). Macroscopically, it can be described that the input data are one-dimensional time series data of m channels.

The multi-channel one-dimensional convolution operation is used to process the input data. The sliding convolution kernels move vertically to extract features from the m channels of the input data. The size of the convolution kernel is $1 \times k \times m$, where m is the number of channels. As shown in Figure 1, there are N types of convolution kernels, corresponding to N feature maps. The operation processing of one type convolution kernel is shown in Figure 3. On each channel of input data, the convolution operation is performed with convolution kernels of size $1 \times k$. The resulting matrix of the dot product of the input data and the convolution kernel is summed, and then a basic term is added. The convolution operation can be calculated as follows:

$$M_j = f\left(\sum_{i=1}^m X_i \times K_i^j + b_j\right), \tag{5}$$

where j and i denote the feature map and the channel, respectively; M is the feature map; m is the number of channels; X denotes the input time series data on a channel; K represents the convolution kernel; b_j means the basic matrices. Note that the parameters of the convolution kernels on the m channels are not shared, and the bias b_j is shared. With the translation of the convolution kernels in the vertical direction, the convolution operation is repeated to obtain feature maps. The input data are mapped into higher-dimensional feature maps by the convolution layer. The feature extraction of the USV sensor data is realized. The primary purpose of the pooling layer is to simplify the computational

complexity of the network by compressing the input. Compared to image data, our data dimension is small. Therefore, no pooling operation is required after the convolution layer. We directly take the feature maps of the convolution layer output as the input of the LSTM layer. Moreover, these feature maps retain the time-series characteristics, making the LSTM prediction better.

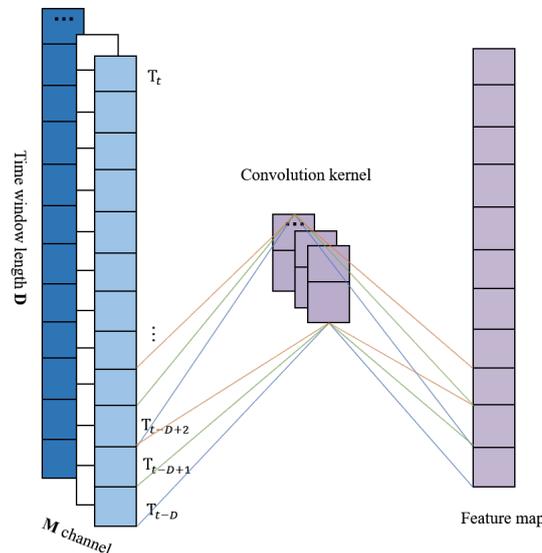


Figure 3. Processing of the M -channel 1D-convolution operation.

2.3. LSTM-Based USV Roll Motion Modeling

LSTM is an elegant variant of the recurrent neural network (RNN) developed by Hochreiter Sepp and Schmidhuber Jürgen [37]. It solves the problems of gradient explosion and gradient disappearance in RNN. The core concept of LSTM is cell state and “gate”. The cell state is equivalent to the path of information transmission, allowing information to pass through the sequence chain, which can be regarded as the “memory” of the network. In theory, the cell state is able to pass on relevant information during the sequence processing. The “gates” are internal mechanisms that regulate the flow of information by removing or adding information to the cell state. The LSTM cell includes a forget gate, input gate, and output gate as depicted in Figure 4, which are composed of sigmoid and tanh activation functions, pointwise multiplication operation, and pointwise addition operation.

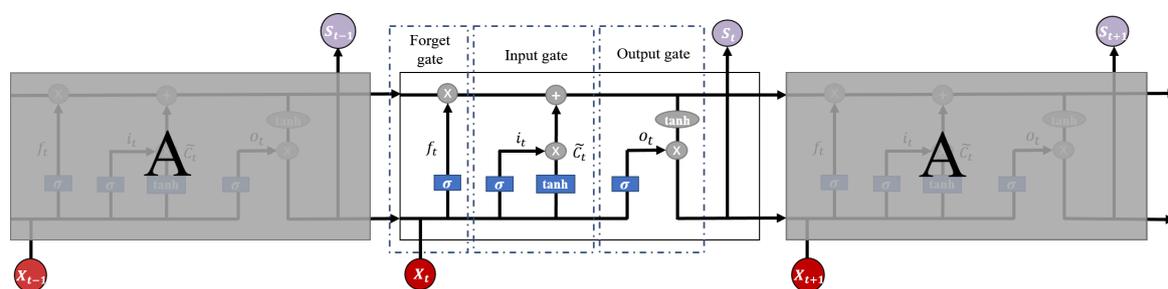


Figure 4. Diagram of an LSTM cell.

In our proposed model, the input of the LSTM layer is the feature maps of the convolution layer output. These feature maps are three-dimensional matrices of $1 \times D - k + 1 \times N$ with time-series characteristics. The third dimension of these matrices represents the features extracted by the convolution layer. The second dimension of these matrices is used as the time step of the LSTM. X_t , S_t , and C_t denote the input, the hidden state, and the cell state at time step t , respectively; W_f , W_i , W_C , W_o , b_f , b_i , b_C , and b_o denote the weight matrices and the basic matrices of the forget gate, the input

gate, the cell state, and the output gate, respectively. The process of USV roll motion modeling based on LSTM is as follows:

- At time step t , the first step is to decide what past step feature maps information will be discarded or retained by the following forget gate f_t :

$$f_t = \sigma(W_f \cdot [S_{t-1}, X_t] + b_f). \tag{6}$$

The information from the previous time step S_{t-1} and X_t are passed to the sigmoid function at time step t . The closer the output value is to 0, the more it will be discarded. The closer it is to 1, the more it will be retained.

- Then, what new information will be stored in the cell state by the input gate i_t is determined. First, the information of the previous time step S_{t-1} and X_t are passed to the sigmoid function to decide what information will be updated. Second, the information of the previous time step S_{t-1} and X_t are passed to the tanh function to create a new candidate value vector \tilde{C}_t . Finally, pointwise multiplication of i_t by \tilde{C}_t is given as the output. The description is as follows:

$$i_t = \sigma(W_i \cdot [S_{t-1}, X_t] + b_i), \tag{7}$$

$$\tilde{C}_t = \tanh(W_c \cdot [S_{t-1}, X_t] + b_c). \tag{8}$$

- The next step is to update the cell state. First, the previous time step C_{t-1} is pointwise multiplied by f_t . This value is then added point by point with the output value of the input gate. This can be given as:

$$C_t = C_{t-1} \otimes f_t \oplus \tilde{C}_t \cdot i_t \tag{9}$$

- The final step is to produce the output by the output gate o_t . First, the information of the previous time step S_{t-1} and X_t are passed to the sigmoid function to determine which parts of cell state will be produced as output. Then, the updated cell state C_t is passed to the tanh function, creating a new candidate value vector. Finally, the output of the tanh function is multiplied pointwise by o_t to calculate S_t . After that, S_t is taken as the output of the current cell, while S_t and C_t are passed to the next time step. This can be described as follows:

$$o_t = \sigma(W_o \cdot [S_{t-1}, X_t] + b_o), \tag{10}$$

$$S_t = o_t \otimes \tanh(C_t). \tag{11}$$

Finally, the fully connected layer connects all the nodes between the adjacent LSTM layer and calculates the final predicted results for USV roll motion at time step $t + 1$.

2.4. Objective Function

In our proposed prediction model, at time step t there are observed values and predicted values, denoted as O_t^r and P_t^r , respectively. The observed values are considered as the true values. The optimization goal is to make the predicted values as close as possible to the true values by back propagation. The loss function can be given as:

$$RMSE = \sqrt{\frac{\sum_i^N (O_i^r - P_i^r)^2}{N}}, \tag{12}$$

where i and N are the subscript of the time step and the prediction duration, respectively. The smaller the value of RMSE is, the better our model. Dropout [42] is used to avoid over-fitting during training.

3. Experiments

In this section, we introduce the process of data collection and pre-processing. The experimental settings and performance indexes of proposed model will be shown in detail. In addition, we provide a brief overview of the reference models.

3.1. Data Set

In this paper, the two data sets employed are the sensor data recorded by “JingHai-VI” and “JingHai-III” USVs [43] during a sea mission in 2018. “JingHai-VI” is a USV for online monitoring of marine environments. “JingHai-III” is mainly used to detect underwater terrain such as island reefs and shallow offshore waters. These sensor data are time-series data, including data recorded by the two USVs during their missions in different waters of China in 2018 with sea state ranging from 1 to 3. The sensor sampling frequency was 5 Hz. Seven pairs of time-series sensor data were used as the data set for the proposed model, including the following features: {roll, pitch, yaw, longitude, latitude, altitude, and speed}. The data of status features: {speed, yaw} contains information about the control system of USV. The data of status features: {roll, pitch, yaw, longitude, latitude, altitude} contain information on the USVs’ six-degrees-of-freedom movement. The sensor data was taken from data collected over the course of a year.

3.2. Data Pre-Processing

First, it is necessary to clean the data because the raw sensor data may contain noisy information. Reducing noise from raw data can minimize its effects on further modeling. The noise source can be either internal or external. In this paper, the major noise source is external because the sensor noise is caused by unavoidable external factors such as gradients and nonhomogeneous media. Statistical estimation was used to remove the internal noise, and the external noise was eliminated by a median filtering technique. Then, the seven pairs of status time-series data were divided into input x and output y by the lag time method. The input x can be expressed as $\{S_{t-D}, S_{t-D+1}, \dots, S_t\}$, and the output y can be expressed as r_{t+1} . Finally, because these data are distributed differently, we normalized them.

3.3. Experimental Settings

The proposed model was implemented with Keras. The set lag time step was 9. The kernel size of the convolution layer was $1 \times 3 \times 7$ and the number of kernels was 128. The hidden units of the two LSTM layers were 64 dimensions. The activation function of all the layers was a ReLU. All weights were constrained by L2 regularization with the weight decay coefficient of 0.0005. Dropout was also applied in all the LSTM layers, with a dropout rate of 0.2. Batch size was set to 32. The model was optimized with Adam with an initial learning rate of 0.001.

3.4. Model Evaluation

In this paper, three performance indexes were used to evaluate the performance of the proposed prediction model. There are root mean square error (RMSE), mean absolute error (MAE), and $M_{2/1}$. The RMSE can be calculated by (1). The calculation formula of MAE is shown below:

$$MAE = \frac{\sum_i^N |O_i^r - P_i^r|}{N}. \quad (13)$$

RMSE and MAE calculate the prediction error of the model. The closer these values are to 0, the better the predictive performance of the model. A more accurate performance improvement between two models can be calculated using the following formula:

$$M_{2/1} = \frac{I_1 - I_2}{I_1} = 1 - \frac{I_2}{I_1}, \quad (14)$$

where $M_{2/1}$ denotes the percentage of performance improvement of model 2 compared to model 1; I_2 and I_1 denote the performance indexes for model 2 and model 1, respectively.

3.5. Reference Models

To demonstrate the advantages of the proposed model, four classic models (i.e., ARIMA, DNN, univariate LSTM, and multivariate LSTM) were compared. The USV roll motion status time-series data were used as the data set for the ARIMA model, DNN, and univariate LSTM. Multivariate LSTM used the same data set as the proposed model.

4. Experimental Results and Discussion

In this section, two real data sets were used to validate the proposed prediction model. We carried out two practical case studies.

4.1. Case Study 1: “JingHai-VI” USV

Several time-series data were selected randomly as the training set from the data set of “JingHai-VI”. These time series data totaled 100,000 time steps. Several representative data were selected as the test sets which were not the part of the training set. These test sets were named “JingHai-VI” test set-1, “JingHai-VI” test set-2, and “JingHai-VI” test set-3, respectively. Test set-1 comprises the data recorded by “JingHai-VI” as it moved in a straight line. Test set-2 comprises the data recorded by “JingHai-VI” in a circular curve motion. Test set-3 is composed of the data recorded by “JingHai-VI” during z-curve movement. There are 2000, 3000, and 5000 time steps in each set, respectively. In Figure 5, panels (a), (b), and (c) are the trajectory history of “JingHai-VI” test set-1, “JingHai-VI” test set-2, and “JingHai-VI” test set-3, respectively. These three trajectories are the most probable for “JingHai-VI” to conduct its mission.

To show the prediction accuracy of the proposed model, the final RMSE and MAE values of each test set are listed in Table 1. Figure 6 shows the part of the final prediction results of our proposed model, where panels (a), (b), and (c) represent the results of “JingHai-VI” test set-1, “JingHai-VI” test set-2, and “JingHai-VI” test set-3, respectively. It can be seen that the predicted values of the proposed model were in good agreement with the actual measured values.

To reveal the prediction performances of ARIMA, DNN, univariate LSTM, multivariate LSTM, and our proposed model, the results of these three test sets are listed in Tables 2–4, respectively. It can be seen that our proposed model had the best performance among all the above. On the whole, the neural network models outperformed the ARIMA model in proving that the linear method is not precise enough to model the nonlinear USV roll motion. The DNN model and univariate LSTM model performed poorly compared to the multivariate LSTM model and the proposed model. They simply model time-series data without considering additional information. Univariate LSTM performed better than DNN due to its ability to deal with long-term sequence data prediction. Although the multivariate LSTM model considers additional information, it does not effectively extract features of this information. Therefore, it did not perform well compared to the proposed model. The decrease percentages $M_{2/1}$ in RMSE of the proposed model compared to the other models are shown in Table 5. Compared to the other models, the performance index RMSE of our proposed model was reduced to varying degrees.

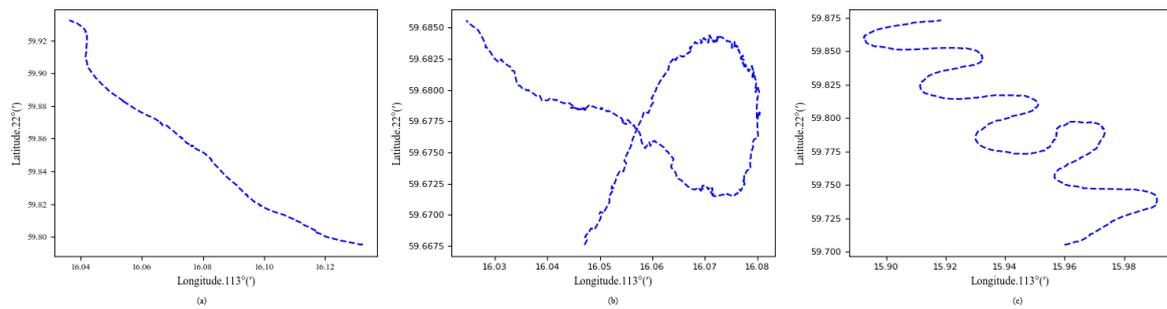


Figure 5. The trajectory history of “JingHai-VI” test sets: (a) test set-1, (b) test set-2, and (c) test set-3.

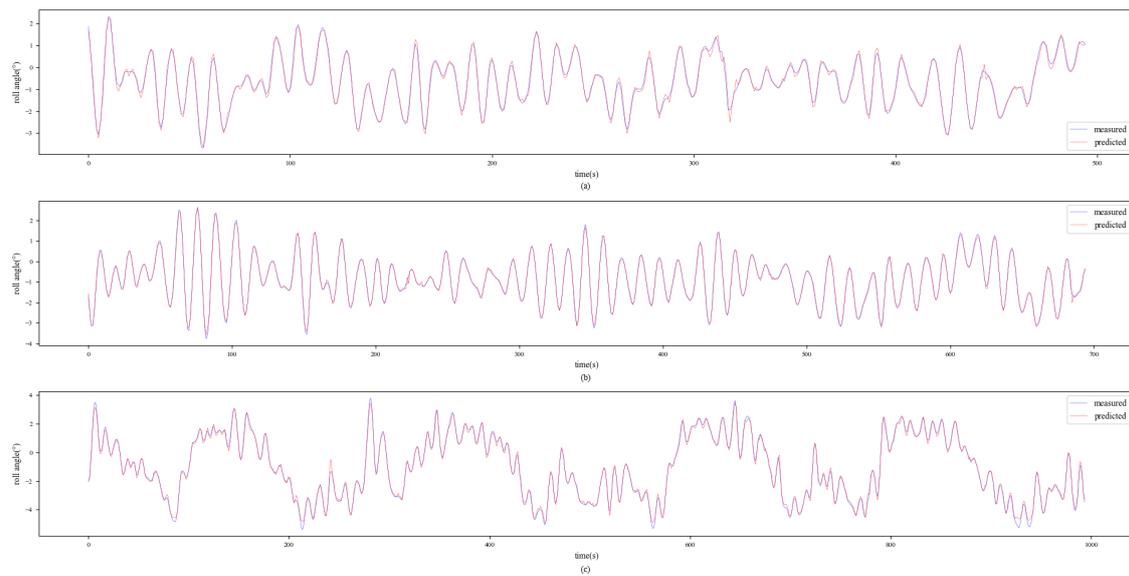


Figure 6. The prediction results of “JingHai-VI” by the proposed model: (a) test set-1; (b) test set-2; and (c) test set-3.

Table 1. Root mean square errors (RMSEs) and mean absolute errors (MAEs) of the proposed model for “JingHai-VI”.

Test Set	RMSE (°)	MAE (°)
“JingHai-VI” test set-1	0.14	0.11
“JingHai-VI” test set-2	0.08	0.06
“JingHai-VI” test set-3	0.16	0.12

Table 2. RMSEs and MAEs of each model at “JingHai-VI” test set-1. ARIMA: autoregressive integrated moving average; DNN: deep neural network.

Model	RMSE (°)	MAE (°)
ARIMA	0.22	0.17
DNN	0.17	0.12
Univariate LSTM	0.16	0.12
Multivariate LSTM	0.15	0.12
CNN+LSTM (proposed)	0.14	0.11

Table 3. RMSEs and MAEs of each model at “JingHai-VI” test set-2.

Model	RMSE (°)	MAE (°)
ARIMA	0.14	0.10
DNN	0.13	0.09
Univariate LSTM	0.12	0.08
Multivariate LSTM	0.09	0.07
CNN+LSTM (proposed)	0.08	0.06

Table 4. RMSEs and MAEs of each model at “JingHai-VI” test set-3.

Model	RMSE (°)	MAE (°)
ARIMA	0.22	0.17
DNN	0.18	0.14
Univariate LSTM	0.16	0.12
Multivariate LSTM	0.18	0.14
CNN+LSTM (proposed)	0.15	0.10

Table 5. Decrease percentage in RMSE of the proposed model compared to other models in each test set of “JingHai-VI”.

Test Set	CNN+LSTM/ARIMA	CNN+LSTM/DNN	CNN+LSTM/Univariate LSTM	CNN+LSTM/Multivariate LSTM
Test set-1	36%	18%	13%	7%
Test set-2	43%	38%	33%	11%
Test set-3	32%	17%	6%	17%

4.2. Case Study 2: “JingHai-III” USV

As in Case Study 1, the 20,000 time-step time-series data were selected as the training set from the dataset of “JingHai-III”. Several representative data were selected as test sets. These test sets were named “JingHai-III” test set-1, “JingHai-III” test set-2, and “JingHai-III” test set-3, respectively. Test set-1 is the data recorded by “JingHai-III” as it moves in a straight line. Test set-2 is the data recorded by “JingHai-III” doing the turning motion. Test set-3 is the data recorded under the curve movement of “JingHai-III”. They were 470, 1400, and 600 time steps in these sets, respectively. Figure 7a–c shows the trajectory history of “JingHai-III” test set-1, “JingHai-III” test set-2, and “JingHai-III” test set-3, respectively. These three trajectories are the most likely to occur when “JingHai-III” conducts a mission.

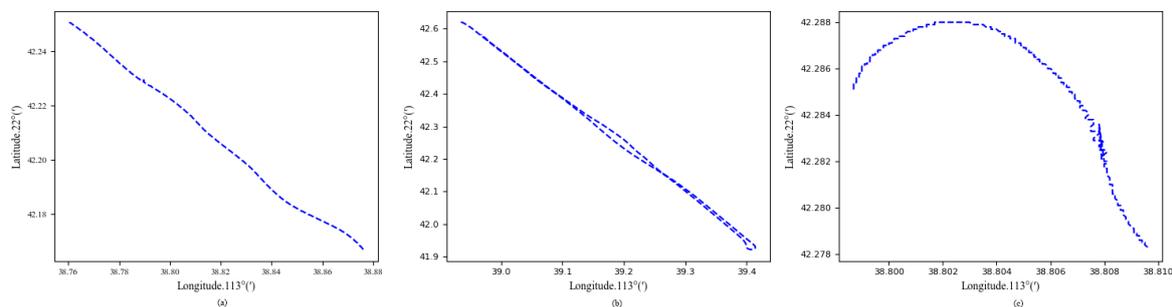


Figure 7. The trajectory history of “JingHai-III” test sets: (a) test set-1; (b) test set-2; and (c) test set-3.

The final RMSE and MAE values of each test set are listed in Table 6. Figure 8 shows that the predicted values fit the actual measured values well. Figure 8a–c shows the final prediction results of the proposed model for “JingHai-III” test set-1, “JingHai-III” test set-2, and “JingHai-III” test set-3, respectively.

The results of these models on three test sets are listed in Tables 7–9, respectively. It can be seen that our proposed model had the best performance. The decrease percentages $M(2/1)$ in RMSE of the proposed model compared to the other models are shown in Table 10. Slightly different from the

discussion of Case Study 1, it was found that the univariate LSTM model performed better than the multivariate LSTM model for the individual test sets. Moreover, the ARIMA model performed well when the USV roll motion changed regularly.

Table 6. RMSEs and MAEs of the proposed model for “JingHai-III”.

Test Set	RMSE (°)	MAE (°)
“JingHai-III” test set-1	0.14	0.10
“JingHai-III” test set-2	0.49	0.36
“JingHai-III” test set-3	0.36	0.22

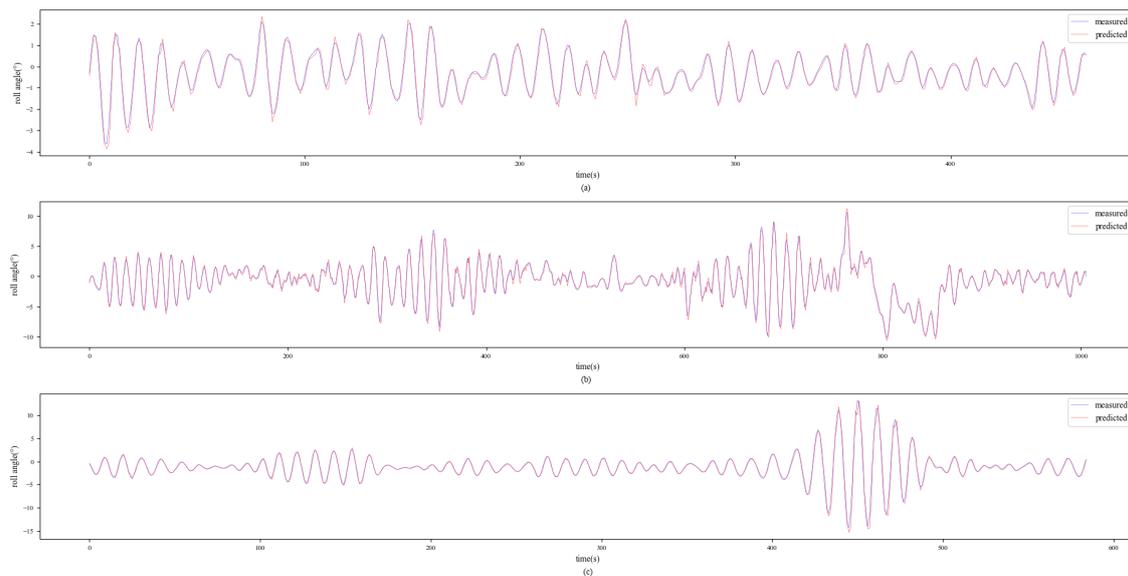


Figure 8. The prediction results of “JingHai-III” by the proposed model: (a) test set-1; (b) test set-2; and (c) test set-3.

Table 7. RMSEs and MAEs of each model in “JingHai-III” test set-1.

Model	RMSE (°)	MAE (°)
ARIMA	0.18	0.12
DNN	0.16	0.12
Univariate LSTM	0.16	0.11
Multivariate LSTM	0.16	0.11
CNN+LSTM (proposed)	0.14	0.10

Table 8. RMSEs and MAEs of each model in “JingHai-III” test set-2.

Model	RMSE (°)	MAE (°)
ARIMA	0.56	0.40
DNN	0.51	0.40
Univariate LSTM	0.53	0.39
Multivariate LSTM	0.51	0.39
CNN+LSTM (proposed)	0.49	0.36

Table 9. RMSEs and MAEs of each model in “JingHai-III” test set-3.

Model	RMSE (°)	MAE (°)
ARIMA	0.37	0.22
DNN	0.38	0.30
Univariate LSTM	0.39	0.28
Multivariate LSTM	0.43	0.31
CNN+LSTM (proposed)	0.36	0.22

Table 10. The decrease percentage in RMSE of the proposed model compared to other models in each test set of “JingHai-III”.

Test Set	CNN+LSTM /ARIMA	CNN+LSTM /DNN	CNN+LSTM /Univariate LSTM	CNN+LSTM /Multivariate LSTM
Test set-1	22%	13%	13%	13%
Test set-2	13%	4%	8%	4%
Test set-3	3%	5%	8%	16%

5. Conclusion and Future Works

In this paper, a coupled CNN and LSTM prediction model is proposed and applied to USV roll motion prediction. The CNN layer extracts spatially relevant and local time-series features of the USV sensor data. The LSTM layer reflects the long-term USV movement process and predicts roll motion for the next moment. The fully connected layer decodes the LSTM output and obtains the final prediction results. Two case studies were carried out. The sensor data measured by “JingHai-VI” and “JingHai-III” USVs of Shanghai University were modeled for roll motion prediction. In both case studies, the experiment results indicated that the proposed model had superior performance to ARIMA, DNN, univariate LSTM, and multivariate LSTM models. We proved that the proposed model was efficient in predicting USV roll motion.

Future work will focus on modeling all six degrees of freedom motion of USVs, and an attention mechanism will be added to the prediction model. Moreover, larger data sets will be used to train the prediction model.

Author Contributions: Conceptualization, W.Z. and P.W.; Resources, Y.P. and D.L.

Funding: This research was funded by the National Science Foundation for Distinguished Young Scholars of P. R. China (under Grant 61525305), State Key Laboratory of Aerodynamics, China Aerodynamics Research and Development Center (under Grant SKLA20180303), Natural Science Foundation of Shanghai (under Grant 19ZR1417700), Project of Shanghai Municipal Science and Technology Commission (under Grant 17DZ1205000), High Performance Computing Center, Shanghai University.

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

References

- Bertram, V. Unmanned surface vehicles—A Survey. In Proceedings of the Skibsteknisk Selskab, Copenhagen, Denmark, March 2008; pp. 1–14.
- Naeem, W.; Sutton, R.; Chudley, J. Modelling and control of an unmanned surface vehicle for environmental monitoring. In Proceedings of the UKACC International Control Conference, Glasgow, UK, 30 August–1 September 2006; pp. 1–6.
- Nikolakopoulos, K.; Lampropoulou, P.; Fakiris, E.; Sardelianos, D.; Papatheodorou, G. Synergistic Use of UAV and USV Data and Petrographic Analyses for the Investigation of Beachrock Formations: A Case Study from Syros Island, Aegean Sea, Greece. *Minerals* **2018**, *8*, 534. [[CrossRef](#)]
- Majohr, J.; Buch, T. Modelling, simulation and control of an autonomous surface marine vehicle for surveying applications Measuring Dolphin MESSIN. *Adv. Unmanned Mar. Veh.* **2006**, *7*, 329–352.

5. Yang, T.; Sun, N.; Chen, H.; Fang, Y. Neural Network-Based Adaptive Antiswing Control of an Underactuated Ship-Mounted Crane With Roll Motions and Input Dead Zones. *IEEE Trans. Neural. Netw. Learn. Syst.* **2019**, *2*. [[CrossRef](#)]
6. Bačkalov, I.; Kalajdžić, M.; Hofman, M. Inland vessel rolling due to severe beam wind: A step towards a realistic model. *Probabilistic Eng. Mech.* **2010**, *25*, 18–25. [[CrossRef](#)]
7. Bulian, G.; Francescutto, A. Effect of roll modelling in beam waves under multi-frequency excitation. *Ocean Eng.* **2011**, *38*, 1448–1463. [[CrossRef](#)]
8. Francescutto, A.; Contento, G. An Investigation On the Applicability of Simplified Mathematical Models to the Roll-Sloshing Problem. *Int. J Offshore. Polar.* **1999**, *8*, 8.
9. van Daalen, E.; Gusing, M.; Grasman, J.; Remmert, J. Roll dynamics of a ship sailing in large amplitude head waves. *J. Eng. Math.* **2014**, *89*, 137–146. [[CrossRef](#)]
10. Silva, S.; Guedes Soares, C. Prediction of parametric rolling in waves with a time domain non-linear strip theory model. *Ocean Eng.* **2013**, *72*, 453–469. [[CrossRef](#)]
11. Huang, L.; Duan, W.; Han, Y.; Chen, Y.S. A review of short-term prediction techniques for ship motions in seaway. *J. Ship Mech.* **2014**, *18*, 1534–1542.
12. Sidar, M.; Doolin, B. On the feasibility of real-time prediction of aircraft carrier motion at sea. *IEEE Trans. Autom. Control* **1983**, *28*, 350–356. [[CrossRef](#)]
13. Triantafyllou, M.; Bodson, M.; Athans, M. Real time estimation of ship motions using Kalman filtering techniques. *IEEE J. Ocean Eng.* **1983**, *8*, 9–20. [[CrossRef](#)]
14. Huang, C.M.; Huang, C.J.; Wang, M.L. A particle swarm optimization to identifying the ARMAX model for short-term load forecasting. *IEEE Trans. Power Syst.* **2005**, *20*, 1126–1133. [[CrossRef](#)]
15. Sakellariou, J.; Fassois, S. Stochastic output error vibration-based damage detection and assessment in structures under earthquake excitation. *J. Sound Vib.* **2006**, *297*, 1048–1067. [[CrossRef](#)]
16. Lee, J.; Jun, C.H. Biclustering of ARMA time series. *J. Zhejiang Univ.—Sci. A* **2010**, *11*, 959–965. [[CrossRef](#)]
17. Yumori, I. Real time prediction of ship response to ocean waves using time series analysis. In Proceedings of the OCEANS 81, Boston, MA, USA, 16–18 September 1981; pp. 1082–1089.
18. Xiren, Z.S.Z. A Nonlinear Method of Extreme Short Time Prediction for Warship Motions at Sea. *J. Harbin Eng. Univ.* **1996**, *17*, 1–7.
19. Shi-qiao, Q.; Wei, W. A hybrid AR-DWT-EMD model for the short-term prediction of nonlinear and non-stationary ship motion. In Proceedings of the 2016 Chinese Control and Decision Conference (CCDC), Haikou, China, 19–20 December 2016; pp. 4042–4047.
20. Suhermi, N.; Prastyo, D.D.; Ali, B. Roll motion prediction using a hybrid deep learning and ARIMA model. *Procedia Comput. Sci.* **2018**, *144*, 251–258. [[CrossRef](#)]
21. Deng, L.; Yu, D. Deep learning: Methods and applications. *Found. Trends Signal Process.* **2014**, *7*, 197–387. [[CrossRef](#)]
22. Bacchi, S.; Oakden-Rayner, L.; Zerner, T.; Kleinig, T.; Patel, S.; Jannes, J. Deep Learning Natural Language Processing Successfully Predicts the Cerebrovascular Cause of Transient Ischemic Attack-Like Presentations. *Stroke* **2019**, *50*, 758–760. [[CrossRef](#)]
23. Yeung, S.; Rinaldo, F.; Jopling, J.; Liu, B.; Mehra, R.; Downing, N.L.; Guo, M.; Bianconi, G.M.; Alahi, A.; Lee, J.; et al. A computer vision system for deep learning-based detection of patient mobilization activities in the ICU. *NPJ Digit. Med.* **2019**, *2*. [[CrossRef](#)]
24. Wang, D.; Devin, C.; Cai, Q.Z.; Yu, F.; Darrell, T. Deep Object-Centric Policies for Autonomous Driving. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8853–8859.
25. Changshun, W.; Weigang, P.; Huang, Z. Unmanned surface vessels path following system based on adaptive RBFNN. 2017 Chinese Automation Congress (CAC). *IEEE Access* **2017**, *7*, 7539–7545.
26. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall PTR: Upper Saddle River, NJ, USA, 1994.
27. Yang, G.; Jie, Q.M.; Tao, N.Q. Prediction of ship motion attitude based on BP network. In Proceedings of the 2017 29th Chinese Control And Decision Conference (CCDC), Chongqing, China, 28–30 May 2017; pp. 1596–1600.
28. Huang, B.G.; Zou, Z.J.; Ding, W.W. Online prediction of ship roll motion based on a coarse and fine tuning fixed grid wavelet network. *Ocean Eng.* **2018**, *160*, 425–437. [[CrossRef](#)]

29. Yin, J.C.; Perakis, A.N.; Wang, N. A real-time ship roll motion prediction using wavelet transform and variable RBF network. *Ocean Eng.* **2018**, *160*, 10–19. [[CrossRef](#)]
30. Yin, J.; Wang, N.; Perakis, A.N. A real-time sequential ship roll prediction scheme based on adaptive sliding data window. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *48*, 2115–2125. [[CrossRef](#)]
31. Yu, Y.; Sheno, R.A.; Zhu, H.; Xia, L. Using wavelet transforms to analyze nonlinear ship rolling and heave-roll coupling. *Ocean Eng.* **2006**, *33*, 912–926. [[CrossRef](#)]
32. Zhou, L.; Chen, F. Stability and bifurcation analysis for a model of a nonlinear coupled pitch–roll ship. *Math. Comput. Simul.* **2008**, *79*, 149–166. [[CrossRef](#)]
33. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)]
34. Li, J.; Li, X.; He, D. A Directed Acyclic Graph Network Combined with CNN and LSTM for Remaining Useful Life Prediction. *IEEE Access* **2019**, *7*, 75464–75475. [[CrossRef](#)]
35. Qin, D.; Yu, J.; Zou, G.; Yong, R.; Zhao, Q.; Zhang, B. A Novel Combined Prediction Scheme Based on CNN and LSTM for Urban PM 2.5 Concentration. *IEEE Access* **2019**, *7*, 20050–20059. [[CrossRef](#)]
36. Woo, J.; Park, J.; Yu, C.; Kim, N. Dynamic model identification of unmanned surface vehicles using deep learning network. *Appl. Ocean Res.* **2018**, *78*, 123–133. [[CrossRef](#)]
37. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
38. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
39. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
40. Genovese, A.; Piuri, V.; Plataniotis, K.N.; Scotti, F. PalmNet: Gabor-PCA Convolutional Networks for Touchless Palmprint Recognition. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 3160–3174. [[CrossRef](#)]
41. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [[CrossRef](#)] [[PubMed](#)]
42. Wierzbicki, M.; Pekarik, G. A meta-analysis of psychotherapy dropout. *Prof. Psychol. Res. Pract.* **1993**, *24*, 190. [[CrossRef](#)]
43. Peng, Y.; Yang, Y.; Cui, J.; Li, X.; Pu, H.; Gu, J.; Xie, S.; Luo, J. Development of the USV 'JingHai-I' and sea trials in the Southern Yellow Sea. *Ocean Eng.* **2017**, *131*, 186–196. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).