Modeling of Coupling Effects in Neural Networks for Ship Motion Prediction

Chong Zhang, Student Member, IEEE, Peida Hu, Rong Zhang, Qiuping Wu, Anlan Yang

Abstract

The prediction of ship motions is vital for ship maneuvering and operations at sea, but the prediction accuracy is seriously affected by nonlinear and time-varying ship dynamics. Although various types of neural networks have been used to overcome these problems, the coupling effects in ship motions lack explicit and effective modeling, thereby limiting the prediction accuracy. To address this challenge, this study proposes a novel neural network structure that can explicitly model the coupling effects while remaining frequency-aware and sequence-aware. In the proposed structure, factorization machines are utilized to model the coupling effects and reduce dimensionality. Wavelet transform is used to convey the frequency information, and the recurrent neural network (RNN) structure is applied for online prediction. Experimental results show that with the proposed structure, the root-mean-squared errors of the prediction of roll, pitch, and heave during the 10 seconds ahead are reduced by 5.24%, 3.71%, and 8.71%, respectively, compared with the values for simple RNN models. This finding verifies the outstanding performance of the proposed method despite its low time and space complexity. This study also proposes a multi-stage training trick that can help improve the training process. With this trick, the proposed structure can be applied to newly proposed RNN models as an improvement, as verified through experiments. The interpretability of the proposed method is revealed via visualization, which indicates that the proposed method can help build white-box neural network models for ship motion prediction.

Index Terms

Factorization machines, interpretable machine learning, ship motion prediction, wavelet transforms.

The authors are with the Department of Precision Instrument, Tsinghua University, Beijing 100084, China. (e-mail: chongzh18@mails.tsinghua.edu.cn, hpd211@mail.tsinghua.edu.cn, rongzh@mail.tsinghua.edu.cn, wuqiuping@mail.tsinghua.edu.cn, yal20@mails.tsinghua.edu.cn)
I. INTRODUCTION

Ship motion prediction is a vital topic in ship maneuvering and ocean engineering [1]. The control performance of ships maneuvered through adaptive state feedback control strategies can be improved by decreasing the prediction errors of the system state [2]. Meanwhile, the planning and control performance of robotic manipulators on seaborne platforms are affected seriously by the accuracy of ship motion prediction [3] [4]. For the deployment and landing of air vehicles on board, reliable prediction of ship motions is urgently required to ensure the stability and effectiveness of the shipboard landing control system [5].

However, accurate ship motion prediction is difficult for two main reasons. First, ship dynamics are time-varying [4] and nonlinear [6]. Time-varying environmental factors, such as waves [7], currents [8], and winds [9], can directly affect ship motions. Second, ship motions are highly coupled [10] and have chaos characteristics [11]. Given these factors, traditional parameter-fixed models, such as those based on Kalman filters [12] [13], and linear models, such as autoregressive (AR) models [5], cannot easily achieve high accuracy.

Neural networks are regarded as a common approach for dealing with the limitations of the above-mentioned methods. The effectiveness of neural networks for time-series prediction tasks in other fields, such as energy efficiency [14], multi-robot systems [15], and autonomous electric vehicles [16], has been validated. Neural networks, such as simple backpropagation neural networks [17], recurrent neural networks [18], radial basis function neural networks [1] [19], echo state networks [20], extreme learning machines [21], and hybrid models [22] [23], have also been widely utilized in ship motion prediction.

Given that the attitude of ships varies quasi-periodically, frequency information is also useful for improving the performance of ship motion prediction methods. Wavelet transform is a multi-resolution analysis method with time-frequency localization [24], and the combination of wavelet transform and neural networks can help build frequency-aware models [25]. Three kinds of combination can be used: 1) to use wavelet functions as activation functions [26] [27], 2) to apply wavelet decomposition directly as a feature engineering tool [28] [29] [30], and 3) to embed wavelet decomposition into neural networks [25].

Despite being frequency-aware, all of the models mentioned above fail to model the coupling effects explicitly. The non-linear activation functions in neural networks may result in the coupling of the inputs, but such black-box coupling is not controllable and reliable; thus, the
model cannot easily learn the coupling effects. Moreover, models with high complexity are
time-consuming and unsuitable for real-time ship motion prediction, but embedding wavelet
decomposition into neural networks by allocating independent parallel neural networks to each
sub-series then fully connecting them lead to considerable extra complexity.

Given that the coupling effects can be expressed in the form of multivariate Taylor series
expansion, and the terms beyond the second order can be omitted [31], the coupling effects can
be modeled by the product of two degrees of freedom (DOFs). Ship motion has six DOFs, and the
number of products is $C_6^2 = 15$ in total. For a frequency-aware model, however, the application of
wavelet decomposition significantly increases the dimensionality of the input space; the number
of products also increases dramatically, leading to the curse of dimensionality [32]. For example,
if we decompose each DOF to six dimensions, we will need $C_{36}^2 = 630$ dimensions to represent
the products.

Thus, we cannot directly import product terms into the input space after wavelet decomposi-
tion. Instead, we use factorization machines (FM) to model the coupling effects. FM models can
factorize the coefficient matrix of the products into a low-dimensional matrix to avoid the curse
of dimensionality [33]. For high-order and nonlinear modeling, we also need to subsume FM
under the neural network framework, resulting in neural factorization machines (NFM) [34]. In
this study, the recurrent neural network (RNN) framework is utilized because RNN models are
particularly good at capturing sequential features and structurally friendly to online prediction
[35].

To model the coupling effects effectively while maintaining frequency and sequence awareness,
this study proposes a neural network structure called multi-resolution recurrent neural factoriza-
tion machines (mRNFM). In mRNFM, wavelet decomposition is used to decompose the initial
inputs to sub-series. FM is utilized to model the coupling effects in low-dimensional space, and
it is embedded into the RNN framework. The mRNFM model is expected to be frequency-aware
and lightweight and able to demonstrate the coupling effects.

The remainder of this paper is organized as follows. The methodology is presented in Section
II. The experimental results of the proposed mRNFM model are shown in Section III, and the
discussion is given in Section IV. The conclusions are presented in Section V.
II. Methodology

The proposed mRNFM model is specially designed to model the coupling effects while staying frequency-aware and sequence-aware for ship motion prediction. The proposed model adopts the 6-DOF ship motion model for physical modeling, improved wavelet decomposition to stay frequency-aware, and FM in the RNN framework to model the coupling effects without the curse of dimensionality. The methodology is described as follows.

A. Ship Motions with Six Degrees of Freedom

A 6-DOF model is often built for the study of ship motions, and the ship is assumed to be a rigid body [31]. The DOFs are roll, pitch, yaw, surge, sway, and heave. The coordinate system moves with the ship. The X-axis is the longitudinal coordinate (positive forward), the Y-axis is the transverse coordinate (positive to starboard), and the Z-axis is determined by the right-hand rule. The origin is the installation position of the inertial navigation system (INS). Roll, pitch, and yaw are the rotational DOFs of the X-axis, Y-axis, and Z-axis, respectively. Similarly, surge, sway, and heave are the translational DOFs of the X-axis, Y-axis, and Z-axis, respectively, as is shown in Fig. 1. Roll is denoted by $\phi$, pitch is denoted by $\theta$, yaw is denoted by $\psi$, surge is denoted by $x$, sway is denoted by $y$, and heave is denoted by $z$.

The values of roll, pitch, and heave directly determine the ship motions. By contrast, the values of yaw, surge, and sway make little sense. For example, a single value of yaw contains little information without knowledge of the directions of waves, winds, and currents. However, the rate of yaw matters because it contains information on motion tendency, from which the neural
network can learn about the environmental factors. The rate of surge and sway also matters, although the values themselves are of little significance. Therefore, in this study, $\dot{x}, \dot{y}, \dot{z}, \varphi, \theta,$ and $\psi$ are used to describe ship motions.

Ship dynamics can be generically expressed as

$$\ddot{d}_i = f_i(d_1, \ldots, d_6), \quad i = 1, \ldots, 6,$$

where $d_i$ refers to the $i$-th DOF and $f_i$ is a complicated function. With the form of a multivariate Taylor series expansion, (1) becomes

$$\ddot{d}_i = w_{0,i} + \sum_{j=1}^{6} w_{j,i}d_j + \sum_{j=1}^{6} \sum_{k=1}^{6} w_{jk,i}d_jd_k + h_i(d),$$

where $w_{0,i}$ is the bias coefficient for the $i$-th DOF, $w_{j,i}$ contains the linear coefficients, $w_{jk,i}$ refers to the second-order coefficients, and $h_i(d)$ pertains to the high-order terms with relatively small values [31]. The second-order coefficients correspond to the coupling effects. These coefficients can be regarded as constants in a short period, because the dynamics of heavy objects, such as ships, change slowly.

B. Multilevel Wavelet Decomposition

Multilevel wavelet decomposition is an approach for numerical and functional analyses, and it is based on discrete wavelet transform (DWT) [36]. In time-series prediction, each prediction utilizes the data in the last sliding window. If the transformation is time-variant, the results of the transformation for adjacent windows will be discontinuous, and the output of the neural network cannot be guaranteed to be continuous. Therefore, time-invariant stationary wavelet transform (SWT) [37] is adopted in this study.

SWT is defined by a low-pass filter $\mathcal{H}$, a high-pass filter $\mathcal{G}$, and an upsampling operator $\mathcal{S}$. The low-pass filter $\mathcal{H}$ is defined by a doubly infinite sequence

$$\{h_n\} = \{\ldots, h_{-1}, h_0, h_1, \ldots\},$$

and the action of $\mathcal{H}$ on a doubly infinite sequence $\{x_n\}$ is defined as

$$\{\mathcal{H}x\}_n = \sum_{k} h_{k-n}x_k.$$
In this study, \( \{h_n\} \) is assumed to be compactly supported so that the summary in (4) is finite. The non-zero elements of \( \{h_n\} \) are \( \{h_{-i}, \ldots, h_{-1}, h_0, \ldots, h_{i+1}\} \) for integer \( i \geq 0 \) in standard SWT. Specifically, for Haar wavelet, the non-zero elements are

\[
h_0 = \frac{\sqrt{2}}{2}, \quad h_1 = -\frac{\sqrt{2}}{2}.
\]  

(5)

Similar to the low-pass filter \( \mathcal{H} \), the high-pass filter \( \mathcal{G} \) is defined as

\[
g_n = (-1)^n h_{1-n}, \quad n \in \mathbb{Z}.
\]  

(6)

\( \mathcal{G} \) is compactly supported when \( \mathcal{H} \) is compactly supported. The support of \( \mathcal{G} \) is the same as the support of \( \mathcal{H} \).

The operator \( \mathcal{S} \) is defined as

\[
(Sx)_{2n} = x_n \quad \text{and} \quad (Sx)_{2n+1} = 0.
\]  

(7)

Filters \( H^{[r]} \) and \( G^{[r]} \) are set to have weights \( S^{[r]}h \) and \( S^{[r]}g \), respectively. Then, the output of standard SWT is determined from the decomposition of sequence \( x \) as follows:

\[
\begin{cases}
A_1 = \mathcal{H}x \\
A_j = \mathcal{H}^{[j-1]}A_{j-1}, j = 2, 3, \ldots, J \\
D_1 = \mathcal{G}x \\
D_j = \mathcal{G}^{[j-1]}A_{j-1}, j = 2, 3, \ldots, J,
\end{cases}
\]  

(8)

where \( J \) is the level of SWT, \( A \) is for approximation, and \( D \) is for the details. For a finite-length sequence, periodic extension is used at the boundaries to perform computation. Assuming that \( \{x_n\} \) is of length \( l \), the lengths of \( A_i \) and \( D_i \) are all \( l \) \((i = 1, 2, \ldots, J)\). Here, the relation \( 2^J | l \) needs to be satisfied. Due to the upsampling operator \( \mathcal{S} \), the energy of sequence \( A_i \) or \( D_i \) is defined as

\[
E_i = \frac{1}{2^l} \sum_{j=1}^{l} v_j^2,
\]  

(9)

where \( \{v_j\} \) pertains to the elements of \( A_i \) or \( D_i \).

For time-series prediction tasks, data in the past are known, but data in the future are unknown. However, in standard SWT, data in the future are involved in the computation, and they are represented by periodic extension, which can cause errors. These are called boundary effects.
The left boundary has boundary effects, but the periodic extension here can be replaced by past data. Therefore, the filters need to be improved by the translation

\[
\begin{align*}
    h'_k &= h_{k+i+1}, \quad k \in \mathbb{Z} \\
    g'_k &= g_{k+i+1}, \quad k \in \mathbb{Z},
\end{align*}
\]

(10)

where \( i+1 \) is the right support of \( \mathcal{H} \) and \( \mathcal{G} \). With the translation, all of the filters have non-positive support so that the boundary effects are translated to the left boundary and then eliminated by known data in the past.

C. Multi-resolution Recurrent Neural Factorization Machines

The mRNFM model can be viewed as a variant of NFM, and NFM utilizes FM for bi-interaction pooling of the embedded features [34]. FM estimates the output \( y \) by linear combination and factorized interaction of the input vector \( x \) as follows:

\[
\hat{y}_{FM}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} v_i^T v_j \cdot x_i x_j,
\]

(11)

where the estimated model parameters are

\[
w_0 \in \mathbb{R}, \quad w \in \mathbb{R}^n, \quad v \in \mathbb{R}^{n \times k}.
\]

(12)
$k \in \mathbb{N}^*$ is a hyperparameter. The model equation and the parameter updates can be computed in linear time $O(kn)$ [33]. Evidently, the expression of FM is consistent with (2).

Although the interaction between different dimensions of $x$ is modeled, FM is still a multi-variate linear model. Hence, NFM adds the deep neural network (DNN) structure to the output of FM for enhanced expressiveness, as shown in Fig. 2 [34].

The mRNFM model is designed specifically for ship motion prediction to improve the prediction accuracy. For ship motion data, RNN is a better choice for capturing sequential features than common DNNs [35]. Thus, mRNFM replaces the DNN structure in NFM with RNN. SWT is also applied before FM for frequency awareness. Moreover, the DNN structure in NFM only works on the bi-interaction term of one FM, but the RNN framework in mRNFM works on all of the terms in (11) for multiple FM. The reason is that mRNFM is designed for ship motion prediction with dense data, whereas the NFM is designed for sparse predictive analytics of big data. The entire structure of mRNFM is shown in Fig. 3.

1) Input Vectors and Sub-series: The input vector sequence \( \{x_t\} \) consists of \( l \) input vectors ending at the \( t \)-th sample. The input vectors are of the same size \( d \), and the sequence can be viewed as \( d \) sub-sequences. These sub-sequences are dealt with \( J \)-level modified SWT, and each
of them has its own sub-series $D_1, \ldots, D_J$ and $A_1, \ldots, A_J$. Elements with boundary effects need to be eliminated; thus, only the last $m$ elements of each sub-series are preserved.

After these SWT procedures, we derive many sub-series of size $m$ and their initial sequences. The informative and non-redundant ones are selected, and the elements in different sequences annotated by the same subscript are combined as a vector. Then, we obtain the high-dimensional sequence $s_{t-m+1}, \ldots, s_t$.

2) Multiple FM Layer: For each vector $s_{t-m+i}, i \in \{1, 2, \ldots, m\}$ of size $S$, $M$ FM works on it, and the corresponding output vector is $z_i \in \mathbb{R}^M$. The relation between $s_{t-m+i}$ and $z_i$ is formulated as

$$z_{i,r} = \text{FM}_r(s_{t-m+i}), \quad r \in \{1, 2, \ldots, M\},$$

and each FM has the same $k$ but different values of $w_0$, $w$, and $v$. The gradient of $z_i$ to parameter $\theta$ is

$$\frac{\partial z_{i,r}}{\partial \theta} = \begin{cases} 
1, & \text{if } \theta = w_{r,0} \\
 s_{t-m+i,f}, & \text{if } \theta = w_{r,f} \\
 s_{t-m+i,f} \cdot \sum_{j=1}^n v_{r,j,g} s_{t-m+i,j}, & \text{if } \theta = w_{r,f,g} \\
 -v_{r,f,g} \cdot s_{t-m+i,f}^2, & \text{if } \theta = v_{r,f,g}
\end{cases}$$

as implemented in libFM [38]. The subscript $r$ refers to the $r$-th FM.

The multiple FM layer can be activated. The activation function is denoted by $\sigma_z(\cdot)$, and then (13) becomes

$$z_i = \sigma_z(\text{FM}(s_{t-m+i})).$$

The gradient is redefined according to the chain rule.

3) Hidden Layers: The hidden layers are all RNN layers, and the sequence $\{z_i\}_m$ is the input of different time steps. For each layer, the output $P^q_i$ is a function of the input $P^q_{i-1}$ and the output of the previous time step:

$$P^q_i = q(P^q_{i-1}, P^q_{i-1}), i \in \{1, 2, \ldots, m\}.$$

For simplification, define $P^0_i = z_i$, and $P^0_0$ is initialized before calculation. $sz(q)$ is set as the length of $P^q_i$. A simple RNN (sRNN) layer is formulated as

$$P^q_i = \sigma_q(W^q_i P^q_{i-1} + Q^q_i P^q_{i-1} + b_q),$$

where $\sigma_q$ is the activation function. The parameters to be estimated are:

$$W^q \in \mathbb{R}^{sz(q) \times sz(q-1)}, \quad Q^q \in \mathbb{R}^{sz(q) \times sz(q)}, \quad b_q \in \mathbb{R}^{sz(q)}.$$
4) **Output Layer**: The output layer and the hidden layers are designed to be homogeneous. For each time step, the output corresponds to the predicted values of the following $n$ steps. The length of the output is set as the prediction length $n$, and the output of the last time step is the predicted result of the model.

5) **Learning**: The mean square error (MSE) loss is taken as the loss function as follows:

$$L(\hat{y}, y) = \frac{1}{n} ||\hat{y} - y||_2^2$$

where $n$ is the length of $\hat{y}$ and $y$.

The RMSProp optimizer [39] is used in this study. The gradients are calculated with the BPTT algorithm [35].

6) **Online Prediction**: The mRNFM structure is friendly to online prediction because of the time-invariant modified SWT without boundary effects and the RNN structure inside. Only one-step calculation is needed for each new prediction, as presented in Fig. 4. Online training can be performed at given frequencies to help model the time-varying dynamics.

**D. Relation to Existing Models**

1) **Relation to NFM and Other FM-related Models**: NFM and other FM-related models are usually used in recommendation systems for sparse data analytics, whereas mRNFM is specially
designed for ship motion prediction. Despite being used in different fields, FM in these models plays the same role: dimensionality reduction. In mRNFM, FM also directly models the coupling effects.

2) Relation to Different RNN Models: mRNFM utilizes the RNN framework, and different RNN models can all work well in mRNFM. In this study, the sRNN model is used for demonstration.

III. EXPERIMENTAL RESULTS

The proposed mRNFM model is applied to deal with the data collected from a ship maneuvering test. The sampling frequency of the data is 10 Hz, and the length of the data is 5 hours. Several statistical parameters are shown in Table I, with the mean values of roll $\phi$, pitch $\theta$, and heave $z$ set to zero.

A. Preprocess

Table I shows that all of the features, except for the rate of surge $\dot{x}$, can be viewed as zero-centered with their range in the same order of magnitude. For a sailing ship, $\dot{x}$ is usually positive because it indicates the sailing velocity. However, for improved performance of the neural network, we need to normalize $\dot{x}$ as follows:

$$\dot{x} \leftarrow \frac{\dot{x} - \text{Mean}(\dot{x})}{\text{Std}(\dot{x})},$$

(20)
Fig. 5. Relative energy of each level of SWT for different features: (a) roll, (b) pitch, (c) yaw rate, (d) surge rate, (e) sway rate, and (f) heave.

where $Mean(\dot{x}) = 6.34$ and $Std(\dot{x}) = 0.357$ according to Table I.

Then, modified SWT is applied to estimate the features, and the choice of level is important for effective and non-redundant representation. For the selection of level $J$, the following criteria need to be satisfied.

1) The high-frequency part that has low energy and little information but much noise should be separated then discarded.

2) The energy of $A_J$ or $D_J$ should be moderate so that the frequency domain is fully separated without too much redundancy.

3) The energy of approximation $A_J$ should be larger than the energy of the details $D_J$.

Haar wavelet [40] is utilized in this study to achieve a simple but effective implementation. For each feature, the energy of $A_i$ and $D_i$ for $i = 1, 2, \ldots, 7$ is computed then divided by the energy of the raw data, as shown in Fig. 5.

According to Fig. 5, level $J = 5$ is a good choice for each feature in consideration of the
TABLE II

NETWORK PARAMETERS

| Model       | Proposed mRNFM | RNN-36 | RNFM-6 | RNN-6 |
|-------------|----------------|--------|--------|-------|
| Input Shape | (256,6)        | (256,6)| (256,6)| (256,6)|
| Layer 0     | SWT            | SWT    | None   | None  |
| Output Shape| (200,36)       | (200,36)| (200,6)| (200,6)|
| Layer 1     | FM \((k = 2)\) | sRNN   | FM \((k = 2)\) | sRNN |
| Activation 1| tanh activation|        |        |       |
| Output Shape| (200,72) for training, (1,72) for each step |
| Layer 2     | sRNN           |        |        |       |
| Activation 2| linear activation|      |        |       |
| Output Shape| (200,36) for training, (1,36) for each step |
| Layer 3     | sRNN           |        |        |       |
| Activation 3| tanh activation|        |        |       |
| Output Shape| (200,64) for training, (1,64) for each step |
| Layer 4     | sRNN           |        |        |       |
| Activation 4| linear activation|      |        |       |
| Output Shape| (200,100) for training, (1,100) for prediction |

criteria above and the symmetry of different DOFs. Moreover, the energy of \( D_1 \) is very low for all of the features, which means that \( D_1 \) should be discarded. As a result, \((A_5, D_2, D_3, D_4, D_5)\) of each feature, together with the features themselves, make up the 36 sub-series in total. These 36 sub-series work as the inputs of the multiple FM layer in mRNFM.

B. Ablation Study

An ablation study is conducted through an experiment to prove the efficiency of mRNFM. The following types of neural networks are implemented:

1) mRNFM: proposed in this study;
2) RNN-36: mRNFM without the multiple FM layer;
3) RNFM-6: mRNFM without the SWT process;
4) RNN-6: mRNFM without the SWT process and the multiple FM layer.
Among the models above, RNN-36 takes the 36 sub-series as the input of a common RNN model, RNFM-6 takes the 6 initial sequences as the sub-series, and RNN-6 takes the 6 initial sequences as the input of a common RNN model. The network parameters are listed in Table II. The length of the input vector sequence is 25.6 s. The length of the sub-series is 20.0 s. The prediction length is 10.0 s. For online prediction, the hidden state in the RNN layers can be reused for each new step, thereby greatly reducing the inference time.

C. Training

1) Splitting the Dataset: The training samples are sampled from the last 70% of the data, with the sampling interval being 0.5 s. The samples for the test are obtained from the first 20% of the data with the same sampling interval so that they are not strongly correlated to the training samples. The remaining data make up the validation set. The models for different DOFs are independently trained as they can run in parallel during the inference.

2) Optimizer: RMSProp optimizer is utilized for training. Learning rate $\alpha$ is 0.002 with decay of 0.02 per epoch. The decay factor of the squared gradients is $\rho = 0.85$.

3) Batch Learning and Early Stopping: The batch size is 32, and the samples are shuffled before each epoch. The patience for early stopping is 4.

D. Metrics and Results

The metrics are the RMSE of the prediction of roll, pitch, and heave during the 10 seconds ahead. The number of trainable parameters in the different models are presented. The results are shown in Table III.
According to the test results, the proposed mRNFM model achieves the best performance, although the number of trainable parameters is slightly larger than that of others. The test results indicate that the combination of multilevel wavelet decomposition and FM provides improved predictions, but using only one of them cannot ensure a small loss.

Fig. 6 shows how the errors of the proposed mRNFM model and the RNN-6 model change with the predicted time point. The errors increase as the predicted time point goes ahead, and the mRNFM model outperforms the RNN-6 model during the entire prediction.

The inference time of the mRNFM model is about 60 ms per sample (200 predictions), indicating that the proposed model is lightweight enough for real-time prediction even with much deeper Neural Network structures. The training time is about 400 ms per batch (32 samples), indicating that online learning can be implemented for real-time ship motion prediction. The runtime environment is as follows: Tensorflow 1.15.2 + Keras 2.3.1 (Python 3.6.9), Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz × 8 (RAM 8 GB).

E. Multi-stage Training Trick

During training, the optimizer may converge to a relatively high loss because the bi-interaction terms in the multiple FM layer can affect the data distribution, which is usually maintained by batch normalization [41] in common DNN models, leading to poor gradients. For instance, if $x \sim N(0, 1)$, $x^2$ does not follow the Gaussian distribution, and it can not be simply dealt with
by reducing the internal covariate shift because the distribution is skewed.

Such a problem can be solved by multi-stage training. According to (11), the multiple FM layer degenerates to a full-connection (FC) layer when the parameter $v$ of each FM is set to zero. At the same time, the gradients of $v$ become zero according to (14), thereby making $v$ remain zero during the training process. After the validation loss converges and before overfitting, we can unfreeze $v$ by assigning relatively small values and continue training to help the model fit better. In brief, the training process is split into two stages: Stage 1 with $v = 0$ and Stage 2 with unfrozen $v$.

Even SOTA models can be improved by replacing the corresponding linear layer with the multiple FM layer and applying such multi-stage training skills. We take the recently proposed multiscale attention-based LSTM (MSA-LSTM) model [28] for demonstration. The structures of the MSA-LSTM model and the mRNFM-modified MSA-LSTM model are presented in Fig. 7. The difference is between the linear layer and the multiple FM layer. The multiscale attention
structure in MSA-LSTM is maintained, but the SWT layer and the output shape are changed to be consistent with the models in the ablation study.

Fig. 8 presents the test error of the three models below for roll prediction.

1) MSA-LSTM model, RMSE = 0.3380°

2) mRNFM-modified MSA-LSTM model applying the multi-stage training trick, RMSE = 0.3345°

3) mRNFM-modified MSA-LSTM model without the multi-stage training trick, RMSE = 0.3391°

Although the mRNFM-modified MSA-LSTM model can outperform the original MSA-LSTM model, the multi-stage training trick is necessary. The advantages of mRNFM-modification appear to be slight because of the leveling effect by the basic MSA-LSTM structure. All of the three models make accurate prediction at $t < 2$ s, and all of them fail to reach high accuracy at $t > 5$ s. However, between 2 and 5 s, the mRNFM-modified model significantly decreases the test loss by 7.12%.

Fig. 9 shows the effect of the multi-stage training trick on the mRNFM-modified MSA-LSTM model for roll prediction. In the last several epochs, the validation loss indicates over-fitting in Stage 1, but Stage 2 still helps improve the model.

IV. DISCUSSION

A. Generalization of the Methodology

The proposed mRNFM model takes full advantage of the modified SWT process and the multiple FM layer to explicitly model the coupling effects while remaining frequency-aware and
lightweight. It can be applied to different kinds of RNN models because the RNN structure is not strictly limited to this work. Thus, whenever a novel model for ship motion prediction is proposed, the mRNFM structure can be a choice to optimize the model. For instance, in this study, the mRNFM structure is applied not only to the sRNN model, but also to the MSA-LSTM model. The multi-stage training trick can also directly optimize the model based on the pretrained parameters.

B. Interpretability: Coupling Effects

As an attempt to explicitly model the coupling effects, mRNFM naturally gains interpretability of the coupling effects. The bi-interaction terms in FM units can reflect the strength of the coupling effects.

We take the mRNFM model implemented in the ablation study for demonstration. For each dimension, 72 2D vectors contribute to the bi-interaction terms. In other words, each dimension pair \((x_i, x_j)\) contains 72 bi-interaction coefficients, and this pair’s strength of coupling effects can be defined as the summation of the 72 coefficients. The strength of different pairs is shown in Fig. 10.

For different models, the strength may change greatly because the objectives of prediction (roll, pitch, or heave) are different. Strong coupling effects may exist for several pairs, but
Fig. 10. Pairs’ strength of coupling effects in three mRNFM models. Each DOF has six dimensions (self, $A_5$, $D_2$, $D_3$, $D_4$, $D_5$) corresponding to six columns/rows in order. The three subfigures correspond to the three models: (a) for roll prediction, (b) for pitch prediction, and (c) for heave prediction.

The predicted objective may be unaffected. Still we can see from Fig. 10 that yaw and surge motions are highly coupled with the other motions. The coupling effects between surge and yaw strongly affect the prediction of pitch and heave, and the coupling effects between surge and sway strongly affect the prediction of heave. The resemblance of the figures in Fig. 10 indicates specific patterns of the coupling effects.

Moreover, aside from coupling DOFs, coupling frequencies can also be observed in Fig. 10. For example, parts $D_2$ and $D_3$ of yaw motion can strongly affect the prediction of roll, and parts $D_2$ and $D_3$ of surge motion can affect the prediction of heave.

C. Time and Space Complexity

For the SWT part, the time complexity is equal to that of $2J$ ($J$ is the level of SWT) times of 1D convolution, and the same applies to the space complexity. Therefore, compared with the complexity of neural networks, the complexity of SWT can be ignored.

For each FM, the computation and parameter updates are both $O(kn)$ [33], where $n$ is the number of sub-series. The space complexity of each FM is $O(kn + n + 1)$ according to (12). For the multiple FM layer, the time complexity is $O(Rkn)$, where $R$ is the number of FM units, and the space complexity is $O(R(k + 1)n)$.

For a sRNN layer with $R$ units and $n$-dimensional input, the time complexity is $O(R^2 + Rn)$, and the space complexity is $O(R^2 + Rn)$. Therefore, the complexity of the multiple FM layer
is close to the complexity of a sRNN layer when \( k = 2, n = 36, \) and \( R = 72 \) (parameters in the ablation study).

V. CONCLUSION

The accuracy of ship motion prediction by traditional neural networks is limited by the coupling effects in ship motions. To solve this problem, this study explicitly models the coupling effects of ship motions in neural networks while remaining frequency-aware and sequence-aware. A novel mRNFM structure that combines multilevel wavelet decomposition, FM, and RNN is proposed for the neural network algorithm. SWT is optimized and applied to decompose the sequences into different sub-series, thereby making the model frequency-aware. FM is utilized to model the bi-interactions of different sub-series with low complexity, thus modeling the coupling effects effectively. The RNN structure is fed by the output of the multiple FM layer so that the model becomes sequence-aware and structurally friendly to online prediction.

The experimental study and discussion prove that the proposed mRNFM structure has the following advantages. First, the proposed method makes accurate predictions with low time and space complexity. Second, the proposed method explicitly and effectively models the coupling effects in ship motions, thereby possessing natural interpretability. Third, the proposed method models the interactions between high-frequency and low-frequency parts of the signals. Lastly, the proposed method can directly improve existing RNN models through the multi-stage training trick.

We hope that this work will encourage researchers to explore and improve the proposed method further in the following aspects.

1) Different wavelet functions may have different effects when utilized in the proposed structure; thus, the choice of the wavelet function is worthy of further study.

2) Aside from sRNN, many other neural networks can be applied to mRNFM, which calls for additional experiments.

3) Future studies can also apply the proposed method to other time series tasks, such as EEG and pose recognition.

ACKNOWLEDGMENT

This work was supported by China’s National Training Program of Innovation and Entrepreneurship for Undergraduates.
REFERENCES

[1] J. Yin, N. Wang, and A. N. Perakis, “A real-time sequential ship roll prediction scheme based on adaptive sliding data window,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 12, pp. 2115–2125, 2017.

[2] H. Ma, H. Liang, Q. Zhou, and C. K. Ahn, “Adaptive dynamic surface control design for uncertain nonlinear strict-feedback systems with unknown control direction and disturbances,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 3, pp. 506–515, 2019.

[3] P. J. From, J. T. Gravdahl, and P. Abbeel, “On the influence of ship motion prediction accuracy on motion planning and control of robotic manipulators on seaborne platforms,” in 2010 IEEE International Conference on Robotics and Automation. IEEE, 2010, pp. 5281–5288.

[4] Y. Qian, D. Hu, Y. Chen, Y. Fang, and Y. Hu, “Adaptive neural network-based tracking control of underactuated offshore ship-to-ship crane systems subject to unknown wave motions disturbances,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021.

[5] Y. Huang, M. Zhu, Z. Zheng, and K. H. Low, “Linear velocity-free visual servoing control for unmanned helicopter landing on a ship with visibility constraint,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021.

[6] J. Du, X. Hu, and Y. Sun, “Adaptive robust nonlinear control design for course tracking of ships subject to external disturbances and input saturation,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 1, pp. 193–202, 2017.

[7] P. Han, G. Li, X. Cheng, S. Skjong, and H. Zhang, “An uncertainty-aware hybrid approach for sea state estimation using ship motion responses,” IEEE Transactions on Industrial Informatics, 2021.

[8] M.-W. Li, J. Geng, W.-C. Hong, and L.-D. Zhang, “Periodogram estimation based on lssvr-ccpso compensation for forecasting ship motion,” Nonlinear Dynamics, vol. 97, no. 4, pp. 2579–2594, 2019.

[9] S. Chunyu, X. Zhang, and G. Zhang, “Nonlinear identification for 4 dof ship maneuvering modeling via full-scale trial data,” IEEE Transactions on Industrial Electronics, 2021.

[10] M. Chen, R. E. Taylor, and Y. S. Choo, “Investigation of the complex dynamics of float-over deck installation based on a coupled heave-roll-pitch impact model,” Ocean Engineering, vol. 137, pp. 262–275, 2017.

[11] M.-W. Li, J. Geng, D.-F. Han, and T.-J. Zheng, “Ship motion prediction using dynamic seasonal rsvsr with phase space reconstruction and the chaos adaptive efficient foa,” Neurocomputing, vol. 174, pp. 661–680, 2016.

[12] Y. Wang, M. Soltani et al., “An attitude heading and reference system for marine satellite tracking antenna,” IEEE Transactions on Industrial Electronics, vol. 64, no. 4, pp. 3095–3104, 2016.

[13] M. Sato and M. Toda, “Adaptive algorithms of tuning and switching kalman and $\mathcal{H}_\infty$ filters and their application to estimation of ship oscillation with time-varying frequencies,” IEEE Transactions on Industrial Electronics, vol. 67, no. 1, pp. 501–511, 2019.

[14] Z. Geng, J. Chen, and Y. Han, “Energy efficiency prediction based on pca-frbf model: A case study of ethylene industries,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 8, pp. 1763–1773, 2017.

[15] Y. Zhou, H. Hu, Y. Liu, S. Lin, and Z. Ding, “A real-time and fully distributed approach to motion planning for multirobot systems,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 12, pp. 2636–2650, 2019.

[16] Y. Li, G. Yin, W. Zhuang, N. Zhang, J. Wang, and K. Geng, “Compensating delays and noises in motion control of autonomous electric vehicles by using deep learning and unscented kalman predictor,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, pp. 1–13, 2018.

[17] A. Khan, C. Bil, and K. E. Marion, “Theory and application of artificial neural networks for the real time prediction
of ship motion,” in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2005, pp. 1064–1069.

[18] M. Gao, G. Shi, and S. Li, “Online prediction of ship behavior with automatic identification system sensor data using bidirectional long short-term memory recurrent neural network,” *Sensors*, vol. 18, no. 12, p. 4211, 2018.

[19] J.-C. Yin, A. N. Perakis, and N. Wang, “A real-time ship roll motion prediction using wavelet transform and variable rbf network,” *Ocean Engineering*, vol. 160, pp. 10–19, 2018.

[20] X. Peng, H. Dong, and B. Zhang, “Echo state network ship motion modeling prediction based on kalman filter,” in 2017 *IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2017, pp. 95–100.

[21] J.-C. Yin, Z.-J. Zou, F. Xu, and N.-N. Wang, “Online ship roll motion prediction based on grey sequential extreme learning machine,” *Neurocomputing*, vol. 129, pp. 168–174, 2014.

[22] R. Skulstad, G. Li, T. I. Fossen, B. Vik, and H. Zhang, “A hybrid approach to motion prediction for ship docking—integration of a neural network model into the ship dynamic model,” *IEEE Transactions on Instrumentation and Measurement*, 2020.

[23] T. Wang, G. Li, L. I. Hatledal, R. Skulstad, V. Æsøy, and H. Zhang, “Incorporating approximate dynamics into data-driven calibrator: A representative model for ship maneuvering prediction,” *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.

[24] I. Daubechies, “The wavelet transform, time-frequency localization and signal analysis,” *IEEE transactions on information theory*, vol. 36, no. 5, pp. 961–1005, 1990.

[25] J. Wang, Z. Wang, J. Li, and J. Wu, “Multilevel wavelet decomposition network for interpretable time series analysis,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2437–2446.

[26] S. A. Billings and H.-L. Wei, “A new class of wavelet networks for nonlinear system identification,” *IEEE Transactions on neural networks*, vol. 16, no. 4, pp. 862–874, 2005.

[27] B.-G. Huang, Z.-J. Zou, and W.-W. Ding, “Online prediction of ship roll motion based on a coarse and fine tuning fixed grid wavelet network,” *Ocean Engineering*, vol. 160, pp. 425–437, 2018.

[28] T. Zhang, X.-Q. Zheng, and M.-X. Liu, “Multiscale attention-based lstm for ship motion prediction,” *Ocean Engineering*, vol. 230, p. 109066, 2021.

[29] C. Shen, Y. Chen, G. Yang, and X. Guan, “Toward hand-dominated activity recognition systems with wristband-interaction behavior analysis,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 7, pp. 2501–2511, 2020.

[30] H. Liu, H.-q. Tian, D.-f. Pan, and Y.-f. Li, “Forecasting models for wind speed using wavelet, wavelet packet, time series and artificial neural networks,” *Applied Energy*, vol. 107, pp. 191–208, 2013.

[31] E. M. Lewandowski, *The dynamics of marine craft: maneuvering and seakeeping*. World scientific, 2004, vol. 22.

[32] J. H. Friedman, “On bias, variance, 0/1—loss, and the curse-of-dimensionality,” *Data mining and knowledge discovery*, vol. 1, no. 1, pp. 55–77, 1997.

[33] S. Rendle, “Factorization machines,” in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 995–1000.

[34] X. He and T.-S. Chua, “Neural factorization machines for sparse predictive analytics,” in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017, pp. 355–364.

[35] R. J. Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks,” *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[36] M. J. Shensa, “The discrete wavelet transform: wedding the a trous and mallat algorithms,” *IEEE Transactions on signal processing*, vol. 40, no. 10, pp. 2464–2482, 1992.
[37] G. P. Nason and B. W. Silverman, *The stationary wavelet transform and some statistical applications*. Springer, 1995, pp. 281–299.

[38] S. Rendle, “Factorization machines with libFM,” *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 57:1–57:22, May 2012.

[39] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[40] Z. R. Struzik and A. Siebes, “The haar wavelet transform in the time series similarity paradigm,” in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 1999, pp. 12–22.

[41] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*. PMLR, 2015, pp. 448–456.