

Article

Ship SINS/CNS Integrated Navigation Aided by LSTM Attitude Forecast

Jun Tang ^{1,2}  and Hongwei Bian ^{1,*}

¹ The Department of Navigation Engineering, Naval University of Engineering, Wuhan 430033, China; tjdegree@163.com

² The Department of Navigation, Dalian Naval Academy, Dalian 116018, China

* Correspondence: bhwdgree@163.com; Tel.: +86-1896398985

Abstract: Under the strong interference of sky background noise, the reliability of celestial navigation system (CNS) measurement will drop sharply, which leads to performance deterioration for ships' strapdown inertial navigation system (SINS)/CNS integrated navigation. To solve this problem, a long short-term memory (LSTM) model is trained to forecast a ship's attitude to detect the attitude provided by the CNS, and the LSTM forecasted attitude can also be used as a backup in case of CNS failure. First, the SINS/CNS integrated model is derived based on an attitude solution of the CNS, which provides more favorable feature data for LSTM learning. Then, the key techniques of LSTM modeling such as dataset construction, LSTM coding method, hyperparameter optimization and training strategy are described in detail. Finally, an experiment is conducted to evaluate the actual performance of the investigated methods. The results show that the LSTM model can accurately forecast a ship's attitude: the horizon reference error is less than 0.5' and the yaw error is less than 0.6', which can provide reliable reference attitude for the SINS when the CNS is invalid.

Keywords: CNS; deep learning; integrated navigation; LSTM; SINS



Citation: Tang, J.; Bian, H. Ship SINS/CNS Integrated Navigation Aided by LSTM Attitude Forecast. *J. Mar. Sci. Eng.* **2024**, *12*, 387. <https://doi.org/10.3390/jmse12030387>

Academic Editor: Mihalis Golias

Received: 2 February 2024

Revised: 20 February 2024

Accepted: 22 February 2024

Published: 23 February 2024



Copyright: © 2024 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The SINS is an autonomous navigation system, with the advantages of strong anti-interference ability, good real-time performance and no geographical restrictions, which has become the main navigation system for all kinds of carriers [1,2]. However, due to the accumulation of position, speed and attitude errors over time, the SINS is difficult to use as an independent high-precision navigation system [3,4]. The CNS has the advantages of the non-accumulation of errors over time, strong autonomy, anti-interference ability and high reliability, but it is vulnerable to being affected by sky background noise and has disadvantages such as a low data rate and weak real-time performance and continuity [5–7]. Modern information-based navigation warfare has seriously threatened all kinds of radio navigation systems represented by satellite navigation systems [8]. The complex battlefield electromagnetic environment requires that navigation system must have strong autonomy and anti-interference ability. The SINS and CNS can not only meet the aforementioned requirements, but also have strong complementarities with each other [9,10]. Therefore, in recent years, SINS/CNS integrated navigation has been widely considered and rapidly developed. With great development potential and broad application prospects, the SINS/CNS has become an important direction of integrated navigation technology development [9].

A ship CNS works on the earth's surface; due to harsh astronomical observation conditions, it will be interfered with by strong sky background noise. Even the high-resolution CNS with a small field of view has to face the problem of a sharp decline of observation accuracy and even observation interruption under the impact of atmospheric turbulence, clouds and other meteorological factors [11,12]. It is an urgent problem for SINS/CNS integrated navigation to keep the CNS in a stable working state and provide reliable navigation reference information for the SINS.

In recent years, recurrent neural networks (RNNs) have been widely used in dealing with time series problems, and they provide the inspiration for the investigation in this article. RNNs evolved from the Hopfield neural network proposed by Saratha Sathasivam in 1982, which embodies the idea that “human cognition is based on past experience and memory” [13,14]. The difference with ordinary Deep Neural Networks [15,16] and Convolutional Neural Networks [17,18] is that RNNs not only take into account the influence of the output of the previous moment on the current moment, but also have the memory of past content. Just like the human brain, they do not process and judge current information independently but make optimal choices based on previous knowledge and experience [19]. Therefore, RNNs have significant applicability in processing sequence data and solving serialization-related problems. At present, RNNs have been successfully applied in many artificial intelligence fields, such as Google voice search, Baidu deep speech recognition and Amazon-related products. However, as the length of the time series becomes longer, the memory ability of the RNN will rapidly deteriorate, resulting in “gradient vanishing” or “gradient explosion” [20,21]. As an evolutionary version of RNNs, long short-term memory (LSTM) adopts a gating mechanism and memory cells, which can flexibly control the retention degree of the historical information of time series and effectively solve the problem of “gradient vanishing” or “gradient explosion” [22–24]. At present, LSTM neural networks are being widely used in the field of navigation. Brandon Wagstaff et al. presented a method to improve the accuracy of a zero velocity-aided inertial navigation system (INS) by replacing the standard zero velocity detector with an LSTM neural network [25]. Edoardo Topini et al. developed an LSTM-based dead reckoning approach to estimate surge and sway body-fixed frame velocities, without canonically employing the DVL sensor [26]. Lv et al. proposed a position correction model based on hybrid gated RNNs that does not need to establish a motion model like typical navigation algorithms, to avoid modeling errors in the navigation process [27]. Guang et al. proposed a method using LSTM to estimate position information based on inertial measurement unit (IMU) data and Global Positioning System (GPS) position information [28].

Ships are affected by external factors such as wind, waves, currents and surges, making their attitude change constantly. However, from the perspective of the time dimension, this change can be described by a cosine curve with periodicity. Therefore, the attitude change in a ship is a typical time series problem, and there is a strong long-term correlation between the sequence information. It is highly suitable for the use of LSTM to forecast the ship’s attitude. With the aforementioned considerations, this article is devoted to investigating a method of ship SINS/CNS integrated navigation aided by the forecasted attitude of LSTM. The contributions of this article are as follows:

- (1) The proposed method expands the application boundary of traditional ship SINS/CNS integrated navigation and effectively improves the robustness of the integrated system;
- (2) The SINS/CNS integrated model is derived based on an attitude solution of the CNS, which provides more favorable feature data for LSTM learning;
- (3) The LSTM neural network is developed to accomplish a high-precision ship attitude forecast.

This article is organized as follows. The derivation of the SINS/CNS integrated model based on the attitude solution is presented in Section 2. Section 3 describes the modeling method of the LSTM neural network in detail. In Section 4, an experiment is conducted to verify the performance of the proposed method and the results are presented with a detailed discussion. Section 5 summarizes the whole article.

2. SINS/CNS Integrated Navigation Based on Attitude Solution

The technical proposal of SINS/CNS integrated navigation based on a CNS attitude solution can be described as follows. First, the asynchronous multi-star vectors are transformed into a same star sensor frame. Then, the ship attitude relative to the inertial frame is calculated based on the transformed and inertial space multi-star vectors. Finally, the mea-

surement model is derived based on the inertial attitude solved by the CNS for information fusion with the SINS.

The inertia-based integrated navigation equation with a Kalman filter is well known in the navigation field; it is not repeated in this article as the focus of this chapter is the CNS solution model.

2.1. Transformation for Asynchronous Multi-Star Vectors

With high optical resolution and strong background noise suppression ability, the small field of view CNS is widely used in carriers near the surface of the earth, such as ships and submarines, in poor celestial observation conditions. The small field of view CNS can track only one star at one time, which means that the multi-star vectors needed for every celestial navigation solution are not observed simultaneously in the same star sensor frame. Therefore, the asynchronous multi-star vectors must be corrected for a CNS solution.

It is assumed that the CNS needs to observe m ($m \geq 3$) stars to complete a navigation solution once. Let \mathbf{R}_j^s ($j = 1, 2, \dots, m$) denote the multi-star vectors in the star sensor frame s , and \mathbf{R}_j^i denote the corresponding multi-star vectors in inertial frame i : the transformation of the two can be expressed as

$$\mathbf{R}_j^i = \mathbf{C}_b^i \mathbf{C}_s^b \mathbf{R}_j^s \tag{1}$$

where b denotes the body frame, \mathbf{C}_b^i is the transformed matrix from i to b , which is also named the inertial attitude matrix and \mathbf{C}_s^b is the transformed matrix from b to s , which is also named the installation error matrix.

In the process of practical celestial navigation, \mathbf{C}_s^b is calibrated in the factory and the calibration residual can be estimated by Kalman filtering (KF), \mathbf{R}_j^s represents the observation vectors changing with time and acquired by the star sensor and \mathbf{R}_j^i represents the constant reference vectors obtained with celestial apparent position calculation [29]. Let the superscript “ \sim ” denote the corresponding variable containing the measurement error, so (1) can be converted to

$$\mathbf{R}_j^i = \tilde{\mathbf{C}}_{b(T_j)}^i \mathbf{C}_s^b \tilde{\mathbf{R}}_j^{s(T_j)} \tag{2}$$

$$\mathbf{R}_j^i = \tilde{\mathbf{C}}_{b(T_m)}^i \mathbf{C}_s^b \tilde{\mathbf{R}}_j^{s(T_m)} \tag{3}$$

where $\tilde{\mathbf{C}}_b^i$ is provided by the SINS, T_j denotes the observation time of any star and T_m denotes the observation time of the last star. Frame b has linear or angular motion relative to the inertial space, so it needs to be labeled with time. Since i is a fixed frame, there is no need to label it with time; \mathbf{C}_s^b is usually treated as a constant matrix, so there is no need to time it either. $\tilde{\mathbf{R}}_j^{s(T_j)}$ represents the multi-star vectors in s corresponding to T_j , while $\tilde{\mathbf{R}}_j^{s(T_m)}$ represents the multi-star vectors in s corresponding to T_m . The transformation of the two can be derived from (2) and (3) as

$$\tilde{\mathbf{R}}_j^{s(T_m)} = (\mathbf{C}_s^b)^T (\tilde{\mathbf{C}}_{b(T_m)}^i)^T \tilde{\mathbf{C}}_{b(T_j)}^i \mathbf{C}_s^b \tilde{\mathbf{R}}_j^{s(T_j)} \tag{4}$$

where the right superscript “ T ” is a matrix transpose operator. According to (4), all the asynchronous multi-star vectors can be unified into the star sensor frame corresponding to T_m .

2.2. Inertial Attitude Solved by CNS

After obtaining the multi-star vectors $\tilde{\mathbf{R}}_j^{s(T_m)}$, (3) can be converted to

$$\mathbf{R}_j^i \approx \mathbf{C}_{s(T_m)}^i \tilde{\mathbf{R}}_j^{s(T_m)} \tag{5}$$

where $C_{s(T_m)}^i$ is the inertial attitude matrix of frame s with respect to frame i at T_m . This can be obtained by the algorithm of the multi-vector attitude solution [29] as

$$C_{s(T_m)}^i \approx J(R_j^i, \tilde{R}_j^{s(T_m)}, w_j) \tag{6}$$

where $J()$ is the optimal estimation function for the multi-vector attitude solution and w_j is the weighting coefficient of each star vector. By correcting the installation error matrix C_s^b , the ship's inertial attitude matrix C_b^i can be described as

$$C_b^i = C_{s(T_m)}^i (C_s^b)^T \tag{7}$$

Remark 1. Compared with the yaw or position solutions of the traditional ship CNS, the proposed CNS model based on an attitude solution will provide more comprehensive navigation information, which is also more suitable for LSTM learning.

2.3. Measurement Model for SINS/CNS

The measurement model base on C_b^i for SINS/CNS can be described as [10]

$$m2rv((\tilde{C}_n^e)^T (C_e^i)^T C_b^i (\tilde{C}_b^n)^T) = \phi - N(\delta P) \tag{8}$$

where $m2rv()$ is the transformed function from the matrix to the rotation vector, e denotes the earth frame and n denotes the navigation frame (geographical frame is used as the navigation frame in this article). C_e^i , named the time matrix, is the transformed matrix from i to e , which is calculated based on the high-precision observed time and can be regarded as an error-free quantity. \tilde{C}_n^e , named the position matrix, is the transformed matrix from e to n . \tilde{C}_b^n , named the attitude matrix, is the transformed matrix from n to b . Both \tilde{C}_n^e and \tilde{C}_b^n are provided by the SINS. ϕ is the misalignment angle and $\delta P = [\delta L \ \delta \lambda \ \delta h]^T$ is the position error vector presented in the form of latitude error, longitude error and height error. The matrix N can be expressed as

$$N = \begin{bmatrix} -1 & 0 & 0 \\ 0 & \cos L & 0 \\ 0 & \sin L & 0 \end{bmatrix} \tag{9}$$

where L is the latitude.

3. LSTM Attitude Forecast Model

The diagram of the technical framework for the LSTM attitude forecast model is summarized in Figure 1. The input set is constructed by the attitude \tilde{C}_b^i output from the SINS independent solution module. The target set is constructed by the attitude \hat{C}_b^i output from the SINS/CNS integrated module. The input and target sets are processed by the LSTM encoder to be converted to a data type suitable for the LSTM model training. Then, the LSTM begins learning to determine the hyperparameters and model parameters through iterative training. If the loss continues to decrease, the training continues; otherwise, the forecast starts. The forecasted result is converted to attitude \hat{C}_b^i by the LSTM decoder. With the forecasted attitude as a reference, the attitude \bar{C}_b^i is detected from the CNS module. If the attitude \bar{C}_b^i is invalid, a fault warning is issued and the CNS attitude \bar{C}_b^i will be replaced with the LSTM forecasted attitude \hat{C}_b^i for information fusion with the SINS; otherwise, the training continues.

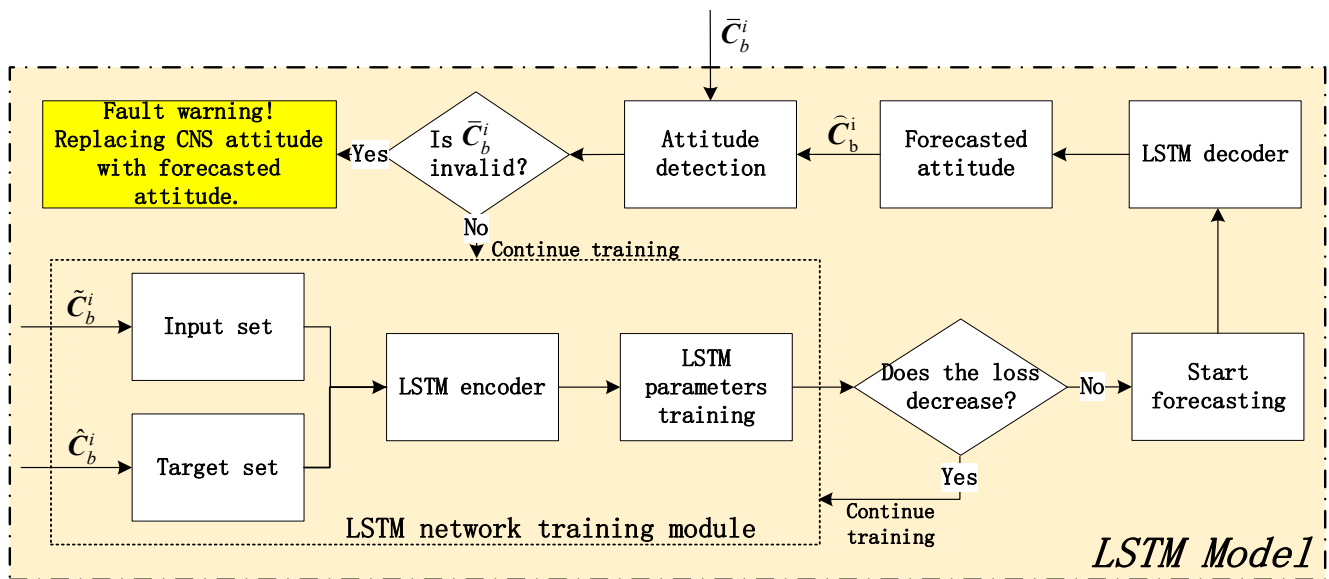


Figure 1. The diagram of technical framework for LSTM attitude forecast model.

Remark 2. In the practice of LSTM model training, the input attitude, the target attitude and the forecasted attitude are all constructed in the form of three-dimensional Euler angles, but in order to facilitate understanding and expression, the attitude is described in the form of a matrix in this article.

3.1. Selection of Input Set and Target Set

The target set refers to the dataset composed of the parameters used as the forecasted target, and the input set refers to the dataset composed of the parameters that can accurately and fully reflect the features of the target. Obviously, the selection of input is determined by the target; whether the input matches the target is the key factor affecting the forecasted performance of the LSTM network.

According to the theory of optimal estimation, the precision of navigation parameters after information fusion with integrated navigation is better than that of each subsystem. Therefore, the attitude \hat{C}_b^i output from the SINS/CNS is selected to construct the target set. The input should not only reflect the attributes of the target attitude, but also have the anti-interference ability without the influence of CNS rejection. Consequently, the SINS independent solution module is added into the SINS/CNS integrated navigation system, and the attitude \tilde{C}_b^i output from this module is selected to construct the input set.

3.2. LSTM Encoder Based on Seq2Seq

The function of the LSTM encoder is to reassemble, segment and normalize the datasets and make them suitable for network training [21]. According to the different constructs of the input and target, the LSTM time series task can be classified as one-to-one, one-to-many, many-to-one and many-to-many [22]. In order to give full play to the advantage of LSTM in processing the long-term sequence, the data structure of Seq2Seq (many-to-many) is used to construct the LSTM encoder.

The method of the second-order overlapping shift is proposed to encode the dataset in this article. The first order means that the target sequence is aligned with the input sequence in the form of an overlapping shift, where the shift step is just the forecasted step (delay). The other order means that the input sequence X and the target sequence Y are divided into a series of subsequences in the form of overlapping shifts.

As shown in Figure 2, L is the length of the input and target sequences, s is the length of the input and target subsequences and d is the forecasted step. With s as the sliding window and 1 as the sliding step, the original input and target sequences are divided into

$L-s + 1$ subsequences, respectively. On this basis, a sample D set is constructed with $L-s + 1$ samples, as shown in Figure 3. The sample set D is divided into a training set, a validation set and a test set according to a certain ratio (usually 6:2:2, 7:2:1 or 8:1:1).

$$\begin{aligned}
 &\text{Input sequence } X: [x_0 \ x_1 \ x_2 \ \cdots \ x_{s-1} \ x_s \ \cdots \ x_{L-1}] \\
 &\text{Target sequence } Y: [y_d \ y_{1+d} \ y_{2+d} \ \cdots \ y_{s+d-1} \ y_{s+d} \ \cdots \ y_{L+d-1}] \\
 &\text{Original input sequence } X: [x_0 \ x_1 \ x_2 \ \cdots \ x_{s-1} \ \cdots \ x_{L-1}] \\
 &\quad \text{Input subsequence } X_1: [x_0 \ x_1 \ x_2 \ \cdots \ x_{s-1}] \\
 &\quad \quad \quad X_2: [x_1 \ x_2 \ x_3 \ \cdots \ x_s] \\
 &\quad \quad \quad X_3: [x_2 \ x_3 \ x_4 \ \cdots \ x_{s+1}] \\
 &\quad \quad \quad \cdots \cdots \cdots \\
 &\quad \quad \quad X_{L-s+1}: [x_{L-s} \ x_{L-s+1} \ \cdots \ x_{L-1}] \\
 &\text{Original target sequence } Y: [y_d \ y_{1+d} \ y_{2+d} \ \cdots \ y_{s+d-1} \ \cdots \ y_{L+d-1}] \\
 &\quad \text{Target subsequence } Y_1: [y_d \ y_{1+d} \ y_{2+d} \ \cdots \ y_{s+d-1}] \\
 &\quad \quad \quad Y_2: [y_{1+d} \ y_{2+d} \ y_{3+d} \ \cdots \ y_{s+d}] \\
 &\quad \quad \quad Y_3: [y_{2+d} \ y_{3+d} \ y_{4+d} \ \cdots \ y_{s+d+1}] \\
 &\quad \quad \quad \cdots \cdots \cdots \\
 &\quad \quad \quad Y_{L-s+1}: [y_{L+d-s} \ \cdots \ y_{L+d-2} \ y_{L+d-1}]
 \end{aligned}$$

Figure 2. The method of second-order overlapping shift.

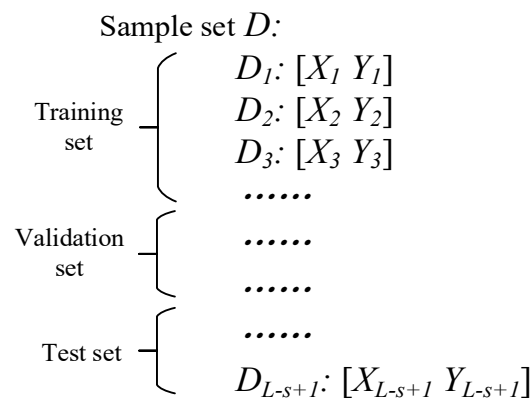


Figure 3. The sample set.

3.3. Bayesian Hyperparameter Optimization

The performance of a deep learning model is mainly determined by two types of parameters, model parameters and model hyperparameters. Model parameters are a set of internal parameters used to determine a deep learning model, such as the weight coefficient matrix and the bias vector in the various gate structures of the LSTM neural network. Hyperparameters are a set of external parameters used to design the network structure and training strategy to optimize the training effect of the model. Model parameters can be obtained independently with deep learning for sample D without manual intervention.

However, the learning of the model parameters is directly affected by the hyperparameters, so the focus is on the problem of hyperparameter optimization as follows.

According to the application characteristics of the ship attitude forecast, the loss function based on the algorithm of mean squared error (MSE) is used as the objective function for hyperparameter optimization.

$$loss(x = x_k) = 1/n \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{10}$$

where x_k represents a set of hyperparameter sampling values, y_i represents the target attitude in the validation set, \hat{y}_i represents the forecasted attitude and n is the number of samples. Hyperparameter optimization aims to find a set of ideal hyperparameters to minimize the objective function. This process can also be simply considered as finding a set of hyperparameters x^* to minimize the attitude forecast error of the LSTM model, as shown in (11).

$$x^* = \operatorname{arccmin} loss(x), x \in X \tag{11}$$

where X is the value range of the hyperparameters to be optimized.

At present, Bayesian optimization has been applied more and more widely in solving the black box function and has become the mainstream method for hyperparameter optimization [30]. Bayesian hyperparameter optimization mainly consists of two technical modules, a learning surrogate model and constructing an acquisition function [31].

3.3.1. Learning Surrogate Model

The objective function belongs to the black box function. It is difficult to obtain the function's specific expression, and the calculation cost is expensive. Therefore, Bayesian optimization must first choose a surrogate model for the objective function. The Gaussian process regression (GPR) model is a widely used and efficient surrogate model. On the basis of assuming that the objective function conforms to the Gaussian distribution, the mathematical expectation $\mu_0(x_{1:k})$ and covariance matrix $\sigma_0(x_{1:k}, x_{1:k})$ are calculated by sampling the input and output k times, and then a Gaussian process regression model $g(x)$ is trained to replace the objective function $loss(x)$. $g(x)$ satisfies the following Gaussian process (GP):

$$g(x) \sim GP(\mu_0(x_{1:k}), \sigma_0(x_{1:k}, x_{1:k})) \tag{12}$$

where

$$\mu_0(x_{1:k}) = [\mu_0(x_1), \mu_0(x_2), \dots, \mu_0(x_k)]^T \tag{13}$$

$$\sigma_0(x_{1:k}, x_{1:k}) = \begin{bmatrix} \sigma_0(x_1, x_1) & \dots & \sigma_0(x_1, x_k) \\ \vdots & \ddots & \vdots \\ \sigma_0(x_k, x_1) & \dots & \sigma_0(x_k, x_k) \end{bmatrix} \tag{14}$$

3.3.2. Constructing Acquisition Function

In order to get the approximate global optimal value of the objective function as soon as possible, it is necessary to construct the acquisition function $a(x|D_n)$ to select the next sampling point x_{n+1} (that is, another set of hyperparameters) optimally, where D_n is expressed as

$$D_n = D_0 \cup \{x_{1:n}, loss(x_{1:n})\} \tag{15}$$

$$D_0 = \{x_{1:k}, loss(x_{1:k})\} \tag{16}$$

where $x_{1:k}$ represents the k sampling points that are used to initialize the surrogate model and $loss(x_{1:k})$ represents the loss values corresponding to $x_{1:k}$. $x_{1:n}$ represents another n sampling points after $x_{1:k}$ and $loss(x_{1:n})$ represents the loss values corresponding to $x_{1:n}$. The surrogate model will continue to learn and update with the subsequent sampling, so

the acquisition function $a(x|D_n)$ can also be considered as the distribution of sampling points based on the given surrogate model, and satisfies

$$a(x|D_n) \sim GP(\mu_n, \sigma_n^2) \tag{17}$$

$$\mu_n = f(x_{1:n}, loss(x_{1:n}), \mu_0(x_{1:k})) \tag{18}$$

$$\sigma_n^2 = m(x_{1:n}, \sigma_0(x_{1:k}, x_{1:k})) \tag{19}$$

where $f()$ and $m()$ are the calculated functions for the mathematical expectation μ_n and covariance matrix σ_n^2 , respectively [32]. Because the explicit expressions of the two functions are relatively complicated, and they can be directly calculated by the Bayesian optimization library, this article does not expand the description.

The expected improvement (EI) model is used to construct the acquisition function. To define the improvement of sampling points

$$I(x_{1:q}) = \max(loss(x^*) - loss(x_{1:q}), 0) \tag{20}$$

where $loss(x^*)$ is the approximate optimal solution of the objective function after the sampling points $x_{1:n}$. $x_{1:q}$ represents the value space of the next sampling point near x^* . If $loss(x_{1:q})$ is smaller than $loss(x^*)$, return the descent degree; otherwise, return 0. Following this law, the acquisition function is constructed as

$$a(x_{1:q}|D_n) = E(I(x_{1:q})) \tag{21}$$

where $E(I(x_{1:q}))$ represents the mathematical expectation of $I(x_{1:q})$. The probability distribution of $a(x|D_n)$ can be determined by (17)–(19), so x_{n+1} is the next sampling point that maximizes the EI of the acquisition function, that is

$$x_{n+1} = \operatorname{argmax} a(x|D_n), x \in x_{1:q} \tag{22}$$

3.3.3. Bayesian Hyperparameter Optimization Algorithm

To sum up, Bayesian hyperparameter optimization can be summarized as Algorithm 1.

Algorithm 1: Bayesian hyperparameter optimization

Input : Initial sampling number k , trial number N , selecting surrogate model (GPR) and acquisition function
Output : $\{x^*, loss(x^*)\}$

- 1 Begin
- 2 Step 1 : Selecting sampling points $x_{1:k}$ randomly to calculate $loss(x_{1:k})$
- 3 Step 2 : Constructing initial set $D_0 = \{x_{1:k}, loss(x_{1:k})\}$, let $n = k$, $D_n = D_0$
- 4 While $n < N$ do
- 5 Step 3 : According to D_n , training surrogate model $g(x)$
- 6 Step 4 : According to the current $g(x)$, determining the distribution of acquisition function $a(x|D_n)$ to obtain $x_{n+1} = \operatorname{argmax} a(x|D_n)$
- 7 Step 5 : According to x_{n+1} , calculating $loss(x_{n+1})$, then updating D_n :
 $D_n = D_n \cup \{x_{n+1}, loss(x_{n+1})\}$
- 8 End
- 9 Output : $\{x^*, loss(x^*)\}$
- 10 End

3.4. Training for LSTM Model

The training for the LSTM model can be summarized as Algorithm 2.

Algorithm 2: Training for LSTM model

```

Input : Training epoch number  $n\_epochs$ , early stop count  $i$ , training set  $D_{tra}$ , validation
set  $D_{val}$ 
Output: LSTM model
1 Begin
2 Step 1: Initializing the default hyperparameters, setting the value space for the
hyperparameters to be optimized
3 Step 2: Instantiating LSTM model
4 While  $epoch < n\_epochs$  do
5 Step 3 : According to  $D_{tra}$ , training and updating surrogate model  $g(x)$ 
6 Step 4 : According to  $D_{val}$ , calculating  $loss_{val}$ 
7 If  $loss_{val}$  does not decrease in consecutive  $i$  epochs of training
8 break, (go to line 11, output the current LSTM model and  $loss_{val}$ )
9 End
10 End
11 Output : LSTM model,  $loss_{val}$ 
12 End
    
```

In fact, the core target of the LSTM model training can be understood as the process of determining the model hyperparameters and model parameters based on the training set and validation set with the loss function as the evaluation index. As mentioned above, Algorithm 2 “training for LSTM model” is closely related to Algorithm 1 “Bayesian hyperparameter optimization”. In practical applications, Algorithm 2 is often nested in Algorithm 1. One of output of Algorithm 2, $loss_{val}$, is just the $loss(x_{1:k})$ in Algorithm 1.

4. Results and Discussion

The LSTM attitude forecast model is established based on the learning of the actual attitude change law of ships, and it works under the supervision of the attitude independently solved by the SINS. Due to the requirements of the relevant policies and confidentiality agreements, the marine observation data of the CNS cannot be disclosed. Therefore, we simulated a ship’s attitude changes at sea through a mathematical model to conduct the experiments for the LSTM attitude forecast and SINS/CNS integrated navigation.

4.1. Experiment Preparation

It is assumed that three stars need to be observed for each celestial navigation solution, and the observed azimuth angle and altitude angle for each star are randomly assigned within the ranges listed in Table 1.

Table 1. Observations of stars.

Star	Rang of Azimuth Angle	Rang of Altitude Angle
S_1^b	$[80^\circ, 100^\circ]$	$[30^\circ, 60^\circ]$
S_2^b	$[-40^\circ, -20^\circ]$	$[30^\circ, 60^\circ]$
S_3^b	$[-160^\circ, -140^\circ]$	$[30^\circ, 60^\circ]$

The projections of the observed stars in inertial frame i can be obtained by

$$S_j^i = C_e^i C_n^e C_b^n S_i^b, \quad i = 1, 2, 3 \tag{23}$$

The performance indexes of the sensors in simulation are listed in Table 2.

Table 2. Performance index of the sensors.

Sensor	Parameter	Index
SINS	Gyro drift	0.03°/h
	Gyro random noise	0.001°/√h
	Accelerometer bias	100 μg
	Accelerometer noise	5 μg/Hz
	Data update frequency	100 Hz
CNS	Measurement noise [azimuth, altitude]	[20'', 10'']
	Observation time for single star	5 s

The initial parameters for simulation are listed in Table 3.

Table 3. The initial parameters.

Parameter	Initial Value
Initial attitude	[0°, 0°, 60°]
Initial velocity	[0.5 m/s, 0.5 m/s, 0 m/s]
Initial position	(30.394491° N, 114.345490° E)
Initial attitude variance	[1', 1', 10'] ²
Initial velocity variance	[0.1 m/s, 0.1 m/s, 0.1 m/s] ²
Initial position variance	[1 m, 1 m, 1 m] ²
Attitude changing amplitude <i>A</i>	[3°, 4°, 2°]
Attitude changing period <i>T</i>	[20 s, 20 s, 18 s]
Attitude initial phase <i>P</i>	[0, π/6, 0]

The changes in the ship’s three-dimensional attitude [*pitch* θ , *roll* γ , *yaw* ψ] are simulated as

$$\theta = A_{\theta} \sin((2\pi / T_{\theta})t + P_{\theta}) \tag{24}$$

$$\gamma = A_{\gamma} \sin((2\pi / T_{\gamma})t + P_{\gamma}) \tag{25}$$

$$\psi = A_{\psi} \sin((2\pi / T_{\psi})t + P_{\psi}) \tag{26}$$

where the amplitude *A*, period *T* and initial phase *P* have been listed in Table 3.

The attitude errors calculated by SINS/CNS integrated navigation are shown in Figure 4.

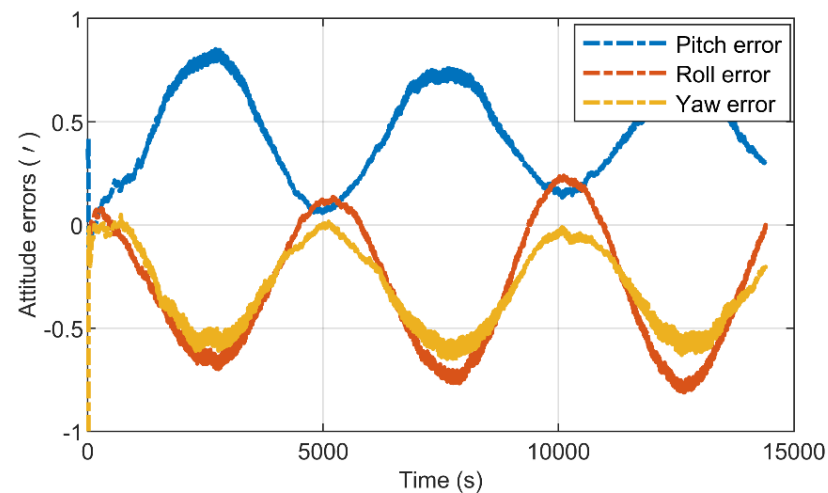


Figure 4. Curves of attitude error.

Remark 3. Since the simulation time is as long as 4 h (14,400 s), the attitude curves along the time axis are too dense to be distinguished; they are not presented in this article.

According to the method described in Section 3.3, the hyperparameters to be optimized and their optimized results are listed in Table 4.

Table 4. The hyperparameters to be optimized.

Parameter	Value Spaces	Optimized Results
Learning rate	$[1 \times 10^{-5} : 1 \times 10^{-2}]$	4.31×10^{-3}
Sequence length	[17, 47, 77, 107]	107
Number of layers	[1, 2, 3, 4]	1
Hidden size	[64, 128, 256, 512]	512
Optimizer	[Adam, RMSprop, SGD]	Adam

Remark 4. Adam (Adaptive Moment Estimation) is a gradient descent algorithm for training neural networks. It combines the momentum algorithm and the adaptive learning rate algorithm to achieve faster convergence and better generalization ability by calculating different adaptive learning rates for each model parameter. In this article, Adam is used to train the weights and the biases of the LSTM. The LSTM model is trained using PyCharm 2021.3.1 with the help of the machine learning library of PyTorch 2.2.0, and the simulation program is implemented using Matlab (<https://ww2.mathworks.cn/en/products/matlab.html>, accessed on 1 February 2023).

4.2. Attitude Forecast

We use the trained LSTM model to forecast the attitude of the SINS/CNS during a period of 13,935 s–14,385 s (450 s). The observation time for each star is 5 s, and it will take 15 s to observe three stars to complete one celestial navigation solution. Therefore, a time series forecast of 30 steps will be performed in 450 s with the single step interval of 15 s. Since the change in the ship’s attitude is much greater than the attitude forecasted error, it is difficult for the attitude tracking curve to distinguish the target attitude from the forecasted attitude. Therefore, the attitude error curve is used to show the forecasted accuracy in Figures 5–7.

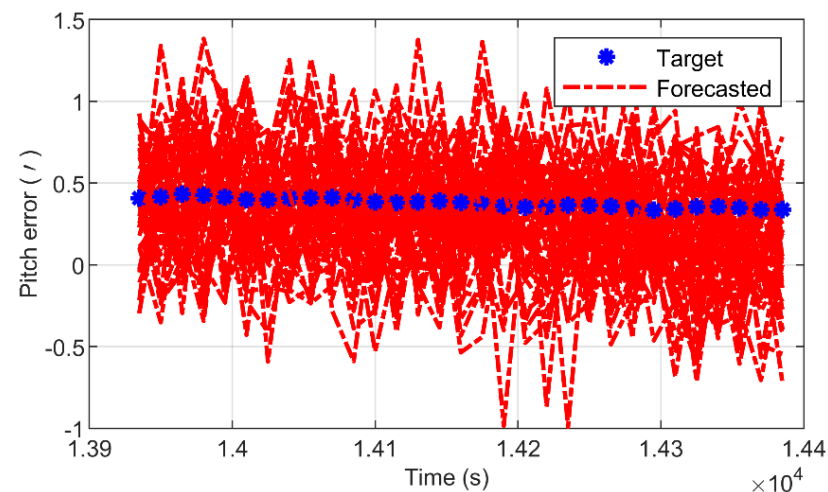


Figure 5. Pitch forecasted error.

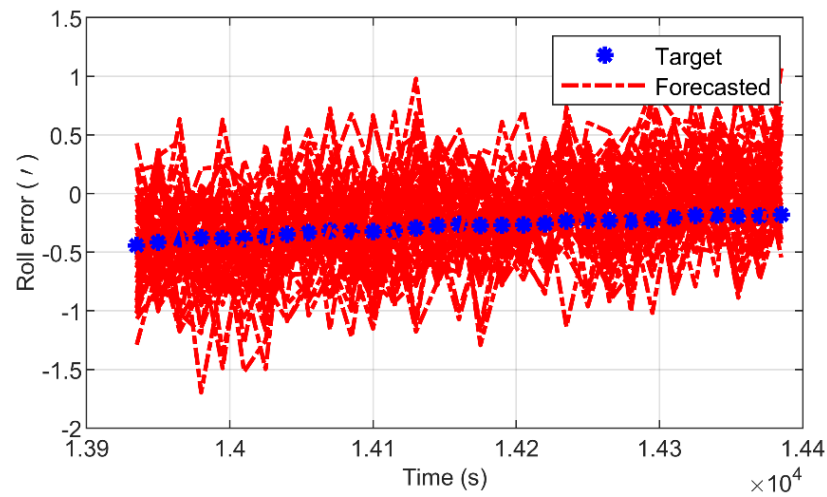


Figure 6. Roll forecasted error.

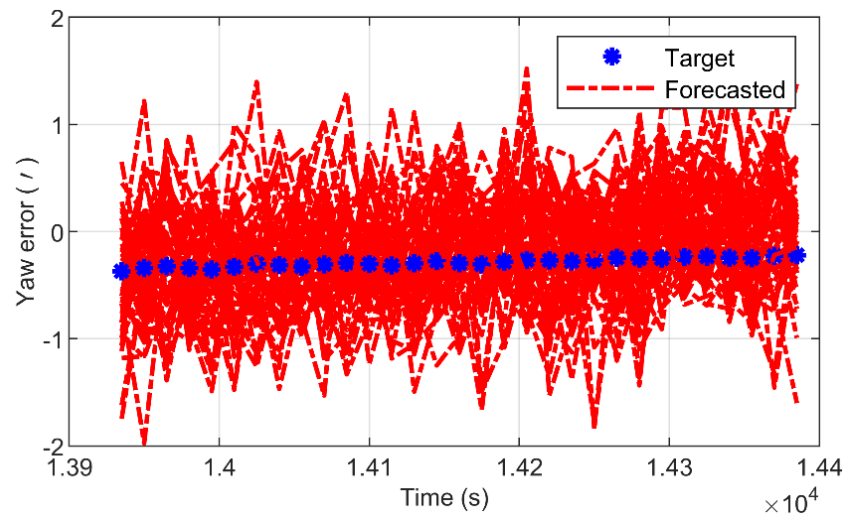


Figure 7. Yaw forecasted error.

To make the results more convincing, 50 Monte Carlo simulation experiments are conducted to evaluate the performance of the LSTM forecast model. The root mean square error (RMSE) is used as a performance metric to quantitatively analyze the forecasted error, and the RMSE of the Monte Carlo simulations is shown in Table 5.

Table 5. RMSE of the forecasted attitude.

Pitch (')	Roll (')	Yaw (')
0.42	0.45	0.57

The RMSE of the forecasted pitch and roll are less than 0.5', and that of the forecasted yaw is less than 0.6'. Although SINS/CNS integrated navigation can effectively suppress the divergence of yaw error, the input of the LSTM model is the attitude independently solved by the SINS, whose yaw error diverges with time, and the forecasted target corresponding to this dimension feature is the yaw provided by SINS/CSN integrated navigation. Therefore, the mapping relationship between the input and output of the yaw is more complicated than that of the pitch or roll, and the forecasted error is also relatively larger. In addition, as can be seen from Figures 5–7, the forecasted results tend to gradually deviate from the target value over time. This is because that although LSTM has excellent

forecasting ability for long time series, the forecasted steps are still limited, as its name, “long short-term memory”, indicates. With the increase in forecasted steps, the reliability of the historical information will gradually weaken, and more and more data excluded in training set will enter into the model, which will result in a larger forecasted error.

4.3. SINS/CNS/LSTM Integrated Navigation

In order to verify the auxiliary effect of the LSTM model on SINS/CNS integrated navigation, the experiment is designed as follows. At a time of 13,935 s in the simulation experiment, 100 times the observation noise (CNS observation noise is shown in Table 2) was injected into the observed star, which led to an abnormal observation value, and then the CNS signal was interrupted.

The experimental results are shown in Figures 8–10, where the blue curve is the attitude error of SINS/CNS integrated navigation during the CNS normal operation. The yellow curve is the attitude error of the SINS independently solved, after the CNS observation is abnormal and interrupted. The red curve is the attitude error of the SINS/LSTM information fusion, after the CNS observation is abnormal and interrupted. Due to the abnormal observation of the CNS, the attitude error changes instantaneously and diverges rapidly with the time update of the SINS. However, the SINS/LSTM information fusion still maintains the propagation trend of the attitude error, which is basically the same as that of SINS/CNS integrated navigation.

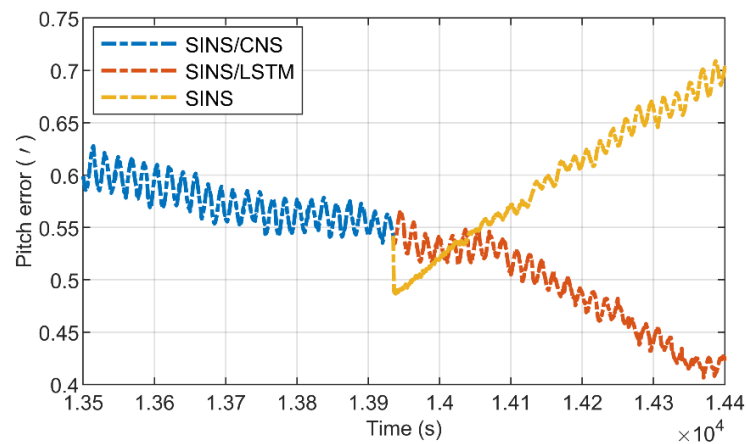


Figure 8. Pitch error based on SINS/CNS/LSTM.

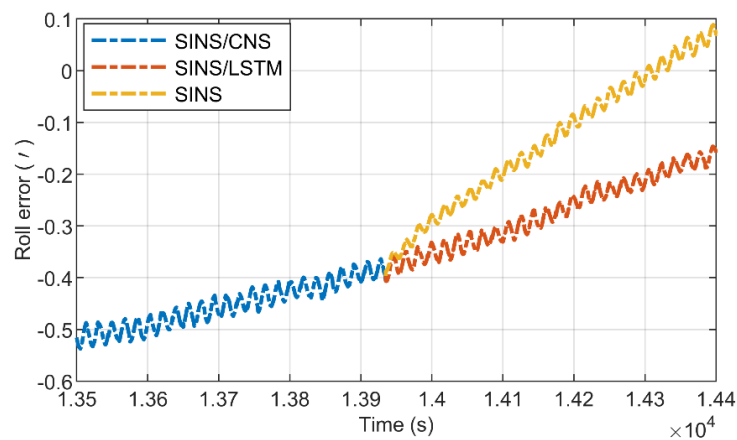


Figure 9. Roll error based on SINS/CNS/LSTM.

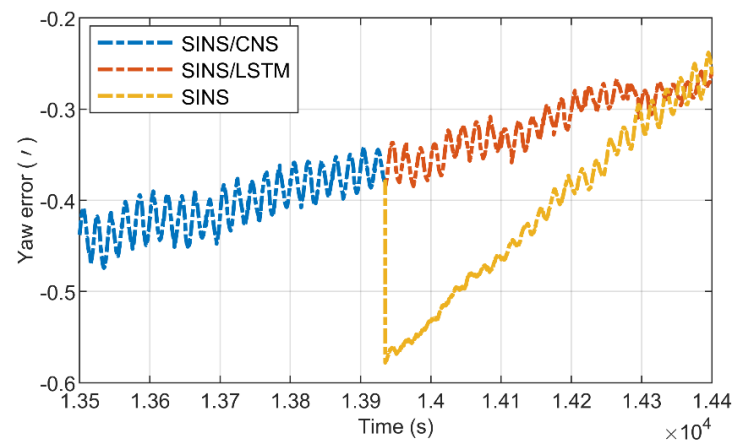


Figure 10. Yaw error based on SINS/CNS/LSTM.

5. Conclusions

This article presented a complete set of technical solutions for ship SINS/CNS integrated navigation aided by LSTM-RNN. The LSTM model accomplished a high-precision attitude forecast for ships. In the case of CNS signal abnormality or even failure, the LSTM forecasted attitude is used as reference information fused with the SINS, which has ensured the continuous and stable output of high-precision navigation information, expanded the application boundary of ship SINS/CNS integrated navigation and effectively improved the robustness of the integrated system.

Compared with the traditional SINS/CNS integrated navigation method, although the technical scheme proposed in this paper increases the computational load of the system through the optimized design of the LSTM network structure and training strategy, the mapping relationship between the input and output of the forecasted model is much simpler. With the support of the integrated computing environment, the increased computational burden is completely acceptable. In addition, in our future research, we will study more accurate and efficient SINS/CNS integrated navigation models that are more suitable for deep learning algorithms.

Author Contributions: J.T.: Conceptualization, Methodology, Software, Writing—Reviewing and Editing, Investigation. H.B.: Supervision, Resources, Funding acquisition, Visualization. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 41876222.

Data Availability Statement: The datasets presented in this article are not readily available due to the requirements of the relevant policies and confidentiality agreement.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Fu, Q.; Wu, F.; Li, S.; Liu, Y. In-motion alignment for a velocity-aided SINS with latitude uncertainty. *IEEE/ASME Trans. Mechatron.* **2020**, *25*, 2893–2903. [[CrossRef](#)]
2. Tang, J.; Bian, H.; Ma, H.; Wang, R. SINS/GNSS Integrated Navigation Based on Invariant Error Models in Inertial Frame. *IEEE Sens. J.* **2024**, *24*, 4290–4303. [[CrossRef](#)]
3. Chang, L.; Li, Y.; Xue, B. Initial alignment for a Doppler velocity log-aided strapdown inertial navigation system with limited information. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 329–338. [[CrossRef](#)]
4. Chang, L.; Hu, B. Robust initial attitude alignment for SINS/DVL. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 2016–2021. [[CrossRef](#)]
5. Qi, Z. Research on the Key Technology of Warship SINS/CNS/DVL Integrated Navigation System. Ph.D. Dissertation, Department of Automation, Harbin Engineering University, Harbin, China, 2015.

6. Li, D.; Wang, Q.; Liu, C.; Zhu, Z. Application of SINS/CNS/DNS integrated navigation in ship navigation system. *J. Nanjing Univ. Inf. Technol. Nat. Sci. Ed.* **2014**, *6*, 321–325.
7. Du, H.; Cao, Y.; Hao, Q.; Yin, H.; Yang, M. The federated filtering algorithm based on INS/GPS/CNS. *Ship Sci. Technol.* **2018**, *40*, 128–131.
8. Tang, J.; Bian, H.; Ma, H.; Wang, R. Initialization of SINS/GNSS Error Covariance Matrix Based on Error States Correlation. *IEEE Access* **2023**, *11*, 94911–94917. [[CrossRef](#)]
9. He, Z.; Qin, S.; Wang, S.; Tan, W.; Liu, D. Star sensor technology applications and future trends in the field of unmanned combat. *J. Ordnance Equip. Eng.* **2016**, *37*, 137–141.
10. Wang, X.L.; Yang, J.; Zhao, Y.N. Introduction. In *SINS/CNS Integrated Navigation Technology*, 1st ed.; Beihang University Press: Beijing, China, 2020; pp. 2–3.
11. Nguyen, V.S.; Im, N.K.; Dao, Q.D. Azimuth method for ship position in celestial navigation. *Int. J. e-Navig. Marit. Econ.* **2017**, *7*, 55–62. [[CrossRef](#)]
12. Zhang, J.L. Key Techniques for Inertial Stellar Integrated Navigation System. Ph.D. Dissertation, Precision Instrument and Machinery, Northwestern Polytechnical University, Xi'an, China, 2015.
13. Sun, L. Research of the SINS/CNS Based on Star Sensor. Ph.D. Dissertation, Department of Automation, Harbin Engineering University, Harbin, China, 2015.
14. Tian, Z.; Zuo, M.J. Health condition prediction of gears using a recurrent neural network approach. *IEEE Trans. Reliab.* **2010**, *59*, 700–705. [[CrossRef](#)]
15. Pan, Y.; Er, M.J.; Li, X.; Yu, H.; Gouriveau, R. Machine health condition prediction via online dynamic fuzzy neural networks. *Eng. Appl. Artif. Intell.* **2014**, *35*, 105–113. [[CrossRef](#)]
16. Li, D.; Wang, W.; Ismail, F. Fuzzy neural network technique for system state forecasting. *IEEE Trans. Cybern.* **2013**, *43*, 1484–1494. [[CrossRef](#)] [[PubMed](#)]
17. Sateesh Babu, G.; Zhao, P.; Li, X.L. Deep convolutional neural network based regression approach for estimation of remaining useful life. In *Database Systems for Advanced Applications, Proceedings of the Database Systems for Advanced Applications: 21st International Conference, DASFAA 2016, Dallas, TX, USA, 16–19 April 2016*; Springer: Cham, Switzerland, 2016; pp. 214–228. [[CrossRef](#)]
18. Li, X.; Ding, Q.; Sun, J.Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliab. Eng. Syst. Saf.* **2018**, *172*, 1–11. [[CrossRef](#)]
19. Yu, W.; Kim, I.Y.; Mechefske, C. An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme. *Reliab. Eng. Syst. Saf.* **2020**, *199*, 106926. [[CrossRef](#)]
20. Chen, J.; Jing, H.; Chang, Y.; Liu, Q. Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process. *Reliab. Eng. Syst. Saf.* **2019**, *185*, 372–382. [[CrossRef](#)]
21. Cheng, Y.; Zhu, H.; Wu, J.; Shao, X. Machine Health Monitoring Using Adaptive Kernel Spectral Clustering and Deep Long Short-Term Memory Recurrent Neural Networks. *IEEE Trans. Ind. Inform.* **2019**, *15*, 987–997. [[CrossRef](#)]
22. Shi, Z.; Chehade, A. A dual-LSTM framework combining change point detection and remaining useful life prediction. *Reliab. Eng. Syst. Saf.* **2021**, *205*, 107257. [[CrossRef](#)]
23. Liu, J.; Lei, F.; Pan, C.; Hu, D.; Zuo, H. Prediction of Remaining Useful Life of Multi-stage Aero-engine Based on Clustering and LSTM Fusion. *Reliab. Eng. Syst. Saf.* **2021**, *214*, 107807. [[CrossRef](#)]
24. Cheng, Y.; Wu, J.; Zhu, H.; Or, S.W.; Shao, X. Remaining Useful Life Prognosis Based on Ensemble Long Short-Term Memory Neural Network. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–12. [[CrossRef](#)]
25. Wagstaff, B.; Kelly, J. LSTM-Based Zero-Velocity Detection for Robust Inertial Navigation. In *Proceedings of the 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Nantes, France, 24–27 September 2018; pp. 1–8. [[CrossRef](#)]
26. Topini, E.; Topini, A.; Franchi, M.; Bucci, A.; Secciani, N.; Ridolfi, A.; Allotta, B. LSTM-based Dead Reckoning Navigation for Autonomous Underwater Vehicles. In *Proceedings of the Global Oceans 2020: Singapore—U.S. Gulf Coast*, Biloxi, MS, USA, 5–30 October 2020; pp. 1–7. [[CrossRef](#)]
27. Lv, P.F.; He, B.; Guo, J. Position Correction Model Based on Gated Hybrid RNN for AUV Navigation. *IEEE Trans. Veh. Technol.* **2021**, *70*, 5648–5657. [[CrossRef](#)]
28. Guang, X.; Gao, Y.; Liu, P.; Li, G. IMU Data and GPS Position Information Direct Fusion Based on LSTM. *Sensors* **2021**, *21*, 2500. [[CrossRef](#)]
29. Tang, J.; Bian, H.W.; Ma, H.; Wang, R.Y. One-step initial alignment algorithm for SINS in the ECI frame based on the inertial attitude measurement of the CNS. *Sensors* **2022**, *22*, 5123. [[CrossRef](#)]
30. Wang, X.; Ren, H.; Guo, X. A novel discrete firefly algorithm for Bayesian network structure learning. *Knowl.-Based Syst.* **2022**, *242*, 108426. [[CrossRef](#)]

31. Han, S.; Liu, X. An extension of multi-attribute group decision making method based on quantum-like Bayesian network considering the interference of beliefs. *Inf. Fusion* **2023**, *95*, 143–162. [[CrossRef](#)]
32. Mosallam, A.; Medjaher, K.; Zerhouni, N. Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction. *J. Intell. Manuf.* **2016**, *27*, 1037–1048. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.