remote sensing

MDPI

Article

Improving Angle-Only Orbit Determination Accuracy for Earth–Moon Libration Orbits Using a Neural-Network-Based Approach

Zhe Zhang [1,2,3,*], Yishuai Shi [1,4] and Zuoxiu Zheng [1,4]

[1] School of Aerospace Engineering, Beijing Institute of Technology, Beijng 100081, China; 3120225032@bit.edu.cn (Y.S.); zhengzx@bit.edu.cn (Z.Z.)
[2] Department of Mathematics and Theories, Peng Cheng Laboratory, No. 2, Xingke 1st Street, Nanshan, Shenzhen 518000, China
[3] Deep Space Exploration Labortory, Beijing 100089, China
[4] Beijing Institute of Technology Chongqing Innovation Center, Chongqing 401120, China
* Correspondence: cnclepzz@126.com

Abstract: In the realm of precision space applications, improving the accuracy of orbit determination (OD) is a crucial and demanding task, primarily because of the presence of measurement noise. To address this issue, a novel machine learning method based on bidirectional long short-term memory (BiLSTM) is proposed in this research. In particular, the proposed method aims to improve the OD accuracy of Earth–Moon Libration orbits with angle-only measurements. The proposed BiLSTM network is designed to detect inaccurate measurements during an OD process, which is achieved by incorporating the least square method (LSM) as a basic estimation approach. The structure, inputs, and outputs of the modified BiLSTM network are meticulously crafted for the detection of inaccurate measurements. Following the detection of inaccurate measurements, a compensating strategy is devised to modify these detection results and thereby reduce their negative impact on OD accuracy. The modified measurements are then used to obtain a more accurate OD solution. The proposed method is applied to solve the OD problem of a 4:1 synodic resonant near-rectilinear halo orbit around the Earth–Moon L2 point. The training results reveal that the bidirectional network structure outperforms the regular unidirectional structures in terms of detection accuracy. Numerical simulations show that the proposed method can reduce the estimated error by approximately 10%. The proposed method holds significant potential for future missions in cislunar space.

Keywords: orbit determination; machine learning; bidirectional long short-term memory; least square method; three-body problem; cislunar space

check for updates

Citation: Zhang, Z.; Shi, Y.; Zheng, Z. Improving Angle-Only Orbit Determination Accuracy for Earth–Moon Libration Orbits Using a Neural-Network-Based Approach. *Remote Sens.* **2024**, *16*, 3287. https://doi.org/10.3390/rs16173287

Academic Editors: Vincenzo Capuano and Roberto Opromolla

Received: 12 August 2024
Revised: 29 August 2024
Accepted: 30 August 2024
Published: 4 September 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 interest in space exploration and the development of space techniques have led to a rapid increase in the number of spacecraft [1–4]. As a result, the need for accurate orbit determination (OD) has become even more crucial for managing long-term space operations [5–8]. OD provides valuable state information that aids in subsequent orbit uncertainty propagation [9–12], guidance [13–15], and control [16–18]. Currently, there are two main groups of state-of-the-art approaches for OD: the Kalman filter (KF) and its variants and the least square method (LSM) and its variants [19–23]. These methods can obtain optimal estimations by combining the state propagation and incoming measurements.

The OD task can be challenging, especially when dealing with short-arc, sparse, and noisy measurements, as well as highly nonlinear dynamics environments [24–26]. To improve the accuracy of OD, various methods have been developed. These methods include the extended Kalman filter (EKF) [27], the unscented Kalman filter (UKF) [28,29], the cubature Kalman filter (CKF) [30–32], and the conjugate unscented Kalman filter

(CUKF) [33,34], all based on the KF algorithm. These methods can capture higher-order terms of the state and measurement equations, which leads to more accurate estimations. However, they also introduce more computational complexity [12,35,36].

The field of space and astrodynamics is witnessing a surge in interest in machine learning approaches, owing to the rapid development of artificial intelligence [37–39]. These approaches provide a promising alternative to enhance the accuracy of orbit prediction and estimation [40–42]. Peng and Bai first proposed an ML-based framework to improve orbit prediction accuracy, in which a support vector machine (SVM) [43] is embedded into a physical-based model to compensate for orbit prediction errors due to the lack of explicit dynamics models. Apart from the SVM, Peng and Bai investigated the artificial neural network (ANN) [44] and Gaussian processes (GPs) [45] for the same purpose, and a comprehensive comparison of these three ML approaches is provided in [46]. Additionally, they proposed a fusion strategy to integrate the ML model with conventional OD methods such as the EKF [47]. These methods are validated to be effective and efficient in simulation-based environments and publicly available data [48]. In addition to the work of Peng and Bai, the boosting tree (BT) and long short-term memory (LSTM) network have also been employed to model the underlying pattern of orbit estimation. Li et al. employed a BT-based method to model the underlying pattern from past OD data and improve the current OD accuracy of space debris under sparse measurements [49]. Zhou et al. proposed a hybrid method for the OD of a continuously maneuvering spacecraft, in which an LSTM network is used to detect and compensate for the maneuver [50,51]. Zhou et al. further proposed a deep neural network (DNN)-based method to compensate for the errors of the low-fidelity dynamics to improve OD accuracy [52].

The proposed method for improving OD accuracy does not rely on modeling the underlying patterns of dynamics like other ML-based approaches. Instead, it focuses on detecting and compensating for inaccuracies in measurements, using a modified bidirectional long short-term memory (BiLSTM) network fused with an LSM. The training data were carefully designed with a focus on input and output structures, with measurements identified as inaccurate if their error is greater than two times the standard deviation (STD). The modified BiLSTM network was then trained to predict which measurements belong to this inaccurate class. One of the advantages of using a BiLSTM network structure is that it can address sequential inputs, making it more adaptable and better at generalizing. Furthermore, the bidirectional structure allows for the use of both past and future information to detect inaccurate measurements at the current epoch, which is not possible with conventional unidirectional LSTM networks. To correct these inaccuracies, a compensation strategy is proposed, and the LSM is used to estimate the orbit. This method is then applied to solve the OD problem of a 4:1 synodic resonant near-rectilinear halo orbit (NRHO) in the Earth–Moon three-body system with promising results.

The remainder of this paper is organized as follows. The essential background and mathematical formulations are given in Section 2. The methodology of the proposed BiLSTM-based machine learning approach to improve OD accuracy is presented in Section 3. Numerical simulations are performed in Section 4, and the conclusions are given in Section 5.

2. Preliminaries

2.1. Least Square Method

First, consider an OD system with a state equation and a measurement equation as follows [53,54]:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{f}[\boldsymbol{x}(t), t] \\ \boldsymbol{z}(t) = \boldsymbol{h}[\boldsymbol{x}(t)] + \boldsymbol{\varepsilon}(t) \end{cases}, \tag{1}$$

where $\boldsymbol{x}(t) = [\boldsymbol{r}(t); \boldsymbol{v}(t)] \in \mathbb{R}^6$ is the state vector of the spacecraft; $\boldsymbol{r}(t) = [x(t), y(t), z(t)]^T \in \mathbb{R}^3$ and $\boldsymbol{v}(t) = [\dot{x}(t), \dot{y}(t), \dot{z}(t)]^T \in \mathbb{R}^3$ represent the position and velocity vectors, respectively; $\boldsymbol{f}[\boldsymbol{x}(t), t] \in \mathbb{R}^6$ is the dynamics model of the spacecraft; $\boldsymbol{z}(t) \in \mathbb{R}^m$ and $\boldsymbol{h}[\boldsymbol{x}(t)] \in \mathbb{R}^m$

denote the measurement vector and the measurement model, respectively; $\boldsymbol{\varepsilon}(t) \in \mathbb{R}^m$ is Gaussian-distributed measurement noise; and m is the dimension of the measurement.

Let $\hat{\boldsymbol{x}}_0 \in \mathbb{R}^6$ be the initial estimated state of the spacecraft at the initial epoch t_0, which can be obtained from an initial OD process or orbit propagation based on a previous OD solution. Then, the estimated orbit $\hat{\boldsymbol{x}}(t) \in \mathbb{R}^6$ at any specified epoch t can be computed by propagating using the dynamics $\boldsymbol{f}[\boldsymbol{x}(t), t]$. Let $\delta\hat{\boldsymbol{x}}_0 \in \mathbb{R}^6$ and $\delta\hat{\boldsymbol{x}}(t) \in \mathbb{R}^6$ be the deviations of the initial estimated state $\hat{\boldsymbol{x}}_0$ and the estimated state $\hat{\boldsymbol{x}}(t)$, respectively. A first-order mapping relationship from $\delta\hat{\boldsymbol{x}}_0$ to $\delta\hat{\boldsymbol{x}}(t)$ is given by

$$\delta\hat{\boldsymbol{x}}(t) \approx \boldsymbol{\phi}_{(t_0,t)}\delta\hat{\boldsymbol{x}}_0\,, \tag{2}$$

where $\boldsymbol{\phi}_{(t_0,t)} = \frac{\partial\hat{\boldsymbol{x}}(t)}{\partial\hat{\boldsymbol{x}}_0}$ represents the state transform matrix (STM). The STM can be obtained by integrating along the estimated orbit $\hat{\boldsymbol{x}}(t)$.

Let $\hat{\boldsymbol{z}}(t) = \boldsymbol{h}[\hat{\boldsymbol{x}}(t)]$ represent the predicted measurement of the estimated state $\hat{\boldsymbol{x}}(t)$. By performing a Taylor series expansion of the measurement model $\boldsymbol{h}[\hat{\boldsymbol{x}}(t)]$ in terms of the estimated state $\hat{\boldsymbol{x}}(t)$, the relationship between the deviation of the predicted measurement $\delta\hat{\boldsymbol{z}}(t)$ and the deviation of the estimated state $\delta\hat{\boldsymbol{x}}(t)$ can be approximated as

$$\delta\hat{\boldsymbol{z}}(t) \approx \boldsymbol{\varphi}_t\delta\hat{\boldsymbol{x}}(t)\,, \tag{3}$$

where $\boldsymbol{\varphi}_t = \frac{\partial\boldsymbol{h}[\hat{\boldsymbol{x}}(t)]}{\partial\hat{\boldsymbol{x}}(t)}$.

Substituting Equation (3) into Equation (2), one has

$$\delta\hat{\boldsymbol{z}}(t) \approx \boldsymbol{\Xi}_{(t_0,t)}\delta\hat{\boldsymbol{x}}_0\,, \tag{4}$$

where $\boldsymbol{\Xi}_{(t_0,t)} \in \mathbb{R}^{m\times 6}$ is the first-order coefficient, given as

$$\boldsymbol{\Xi}_{(t_0,t)} = \boldsymbol{\varphi}_t\boldsymbol{\phi}_{(t_0,t)} \tag{5}$$

Assume that n measurements $\boldsymbol{z}_k = \boldsymbol{h}[\boldsymbol{x}(t_k)] + \boldsymbol{\varepsilon}_k$ $(k = 1, \cdots, n)$ are available, at known epochs t_k, where $\boldsymbol{\varepsilon}_k \in \mathbb{R}^m$ is the measurement noise. The residual between the true measurement $\boldsymbol{z}_k$ and the predicted measurement $\hat{\boldsymbol{z}}(t_k)$ can be written as

$$\begin{aligned}\Delta\boldsymbol{Z} &= \begin{bmatrix} \boldsymbol{z}_1 - \hat{\boldsymbol{z}}(t_1) \\ \vdots \\ \boldsymbol{z}_n - \hat{\boldsymbol{z}}(t_n) \end{bmatrix} = \delta\boldsymbol{h} + \boldsymbol{\varepsilon} \\ &= \begin{bmatrix} \boldsymbol{h}[\boldsymbol{x}(t_1)] - \boldsymbol{h}[\hat{\boldsymbol{x}}(t_1)] \\ \vdots \\ \boldsymbol{h}[\boldsymbol{x}(t_n)] - \boldsymbol{h}[\hat{\boldsymbol{x}}(t_n)] \end{bmatrix} + \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \vdots \\ \boldsymbol{\varepsilon}_n \end{bmatrix} \in \mathbb{R}^{nm}\end{aligned}\,. \tag{6}$$

After substituting Equation (4) into Equation (6), $\delta\boldsymbol{h}$ can be expressed as

$$\delta\boldsymbol{h} \approx \begin{bmatrix} \boldsymbol{\Xi}_{(t_0,t_1)}\delta\hat{\boldsymbol{x}}_0 \\ \vdots \\ \boldsymbol{\Xi}_{(t_0,t_n)}\delta\hat{\boldsymbol{x}}_0 \end{bmatrix}, \tag{7}$$

where $\delta\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0$.

Let $\boldsymbol{\Theta} = \left[\boldsymbol{\Xi}_{(t_0,t_1)}; \cdots; \boldsymbol{\Xi}_{(t_0,t_n)}\right] \in \mathbb{R}^{nm\times 6}$, which, after substituting into Equation (7), Equation (7) can be rewritten as

$$\delta\boldsymbol{h} \approx \Theta\delta\hat{\boldsymbol{x}}_0\,. \tag{8}$$

The goal of the LSM is to obtain an optimal estimation $\delta\hat{\boldsymbol{x}}_0$ that minimizes the residual between the true and predicted measurements [55]. Therefore, the performance metric is written as

$$\begin{aligned} J &= (\Delta\boldsymbol{Z} - \boldsymbol{\Theta}\delta\hat{\boldsymbol{x}}_0)^T(\Delta\boldsymbol{Z} - \boldsymbol{\Theta}\delta\hat{\boldsymbol{x}}_0) \\ &= \mathrm{Tr}\Big((\Delta\boldsymbol{Z} - \boldsymbol{\Theta}\delta\hat{\boldsymbol{x}}_0)(\Delta\boldsymbol{Z} - \boldsymbol{\Theta}\delta\hat{\boldsymbol{x}}_0)^T\Big) \end{aligned}. \tag{9}$$

The optimal estimation $\delta\hat{\boldsymbol{x}}_0$ can be computed by solving the equation $\frac{\partial J}{\partial \delta\hat{\boldsymbol{x}}_0} = \boldsymbol{0}_{1\times 6}$ given by

$$\delta\hat{\boldsymbol{x}}_0 = \Big(\boldsymbol{\Theta}^T\boldsymbol{\Theta}\Big)^{-1}\boldsymbol{\Theta}^T\Delta\boldsymbol{Z}, \tag{10}$$

and the initial estimated state is renewed as $\hat{\boldsymbol{x}}_0 \leftarrow \hat{\boldsymbol{x}}_0 + \delta\hat{\boldsymbol{x}}_0$.

Notice that the optimal estimation $\delta\hat{\boldsymbol{x}}_0$ from Equation (10) is not sufficiently accurate, as only the linear terms are considered during the derivation process (i.e., Equations (2) and (3)). Thus, iterations are needed to make the estimation converge to a specified accuracy. The pseudocode of the LSM is shown as Algorithm 1. The inputs of the algorithm are the dynamics and measurement models, income measurements, initial estimated state $\hat{\boldsymbol{x}}_0$, maximal iteration $i_{\max}$, and threshold η. When the Frobenius norm of the optimal estimation $\delta\hat{\boldsymbol{x}}_0$ is smaller than the given threshold, i.e., $\|\delta\hat{\boldsymbol{x}}_0\| \leq \eta$, the LSM is considered converged. In this work, the threshold is set to 10^{-6}.

Algorithm 1: Pseudocode of least squares method [55]

Input: dynamics model $\boldsymbol{f}[\boldsymbol{x}(t), t]$, measurement model $\boldsymbol{h}[\boldsymbol{x}(t)]$, n measurements z_k, initial estimated state $\hat{\boldsymbol{x}}_0$, maximal iteration $i_{\max}$, threshold η

Output: converged initial estimated state $\hat{\boldsymbol{x}}_0$

$tol \leftarrow \inf$, $i \leftarrow 0$;
while $tol \geq \eta$ and $i \leq i_{\max}$ **do**
  Propagate the estimated orbit $\hat{\boldsymbol{x}}(t)$ using the dynamics $\boldsymbol{f}[\boldsymbol{x}(t), t]$ and the initial estimated state $\hat{\boldsymbol{x}}_0$;
  Compute the predicted measurement $\hat{\boldsymbol{z}}(t_k)$ based on the estimated orbit $\hat{\boldsymbol{x}}(t)$;
  Integrate the STM $\boldsymbol{\phi}_{(t_0,t)}$ and compute the first-order coefficient $\boldsymbol{\Xi}_{(t_0,t)}$ using Equation (5);
  Compute the optimal estimation $\delta\hat{\boldsymbol{x}}_0$ using Equation (10);
  $\hat{\boldsymbol{x}}_0 \leftarrow \hat{\boldsymbol{x}}_0 + \delta\hat{\boldsymbol{x}}_0$;
  $tol \leftarrow \|\delta\hat{\boldsymbol{x}}_0\|$;
  $i \leftarrow i + 1$;
end

2.2. Networks For Addressing Sequential Inputs

The OD is essentially a sequential input problem, as a single incoming measurement is usually insufficient to determine the orbit state. In this subsection, three networks, i.e., the recurrent networks (RNN), LSTM and BiLSTM, that can address sequential inputs are introduced.

2.2.1. RNN

RNN is an elegant way to deal with (epoch) sequential inputs [56,57]. In contrast to conventional machine learning methods, such as SVM [43], ANN [44], GPs [45], and BT [49], which require a fixed number of inputs, the RNN architecture can make use of all the available input information up to the current epoch to predict the current label [58,59]. However, the standard RNN suffers from some drawbacks, such as gradient explosion and disappearance, and not retaining long-distance information [60].

2.2.2. LSTM

The LSTM network is an improved network model extended from the RNN [60,61]. By adding a gating mechanism in the LSTM structure, the drawbacks of the RNN are solved to a certain extent. A standard LSTM structure is shown in Figure 1, in which $\boldsymbol{u}_k$, $\boldsymbol{h}_k$, and $\boldsymbol{C}_k$ represent the input state, the hidden state, and the cell state of the k-th LSTM cell (corresponding to epoch t_k), respectively. The hidden state $\boldsymbol{h}_k$ and the cell state $\boldsymbol{C}_k$ are used to control and store the long-term information of the input state $\boldsymbol{u}_k$. At any specified epoch t_k, the inputs of the k-th LSTM cell include the current input state $\boldsymbol{u}_k$, the previous hidden state $\boldsymbol{h}_{k-1}$ and the previous cell state $\boldsymbol{C}_{k-1}$, whereas the outputs are the current hidden state $\boldsymbol{h}_k$ and the current cell state $\boldsymbol{C}_k$. The mathematical model of the k-th LSTM cell is given as

$$\begin{aligned}
\boldsymbol{f}_k &= \text{sigmoid}\left(\boldsymbol{W}_{fx} \cdot \boldsymbol{u}_k + \boldsymbol{W}_{fh} \cdot \boldsymbol{h}_{k-1} + \boldsymbol{b}_f\right) \\
\boldsymbol{i}_k &= \text{sigmoid}(\boldsymbol{W}_{ix} \cdot \boldsymbol{u}_k + \boldsymbol{W}_{ih} \cdot \boldsymbol{h}_{k-1} + \boldsymbol{b}_i) \\
\boldsymbol{o}_k &= \text{sigmoid}(\boldsymbol{W}_{ox} \cdot \boldsymbol{u}_k + \boldsymbol{W}_{oh} \cdot \boldsymbol{h}_{k-1} + \boldsymbol{b}_o) \\
\tilde{\boldsymbol{C}}_k &= \tanh(\boldsymbol{W}_{cx} \cdot \boldsymbol{u}_k + \boldsymbol{W}_{ch} \cdot \boldsymbol{h}_{k-1} + \boldsymbol{b}_c) \\
\boldsymbol{C}_k &= \boldsymbol{f}_k \otimes \boldsymbol{C}_{k-1} + \boldsymbol{i}_k \otimes \tilde{\boldsymbol{C}}_k \\
\boldsymbol{h}_k &= \boldsymbol{o}_k \otimes \tanh(\boldsymbol{C}_k)
\end{aligned}, \tag{11}$$

where $\boldsymbol{W}_{(\cdot)}$ and $\boldsymbol{b}_{(\cdot)}$ are the weight and bias, respectively, $\text{sigmoid}(\cdot)$ (i.e., $\sigma(\cdot)$ in Figure 1) and $\tanh(\cdot)$ represent the sigmoid and tanh activation functions, respectively, and $\otimes$ denotes the elementwise vector multiplication.

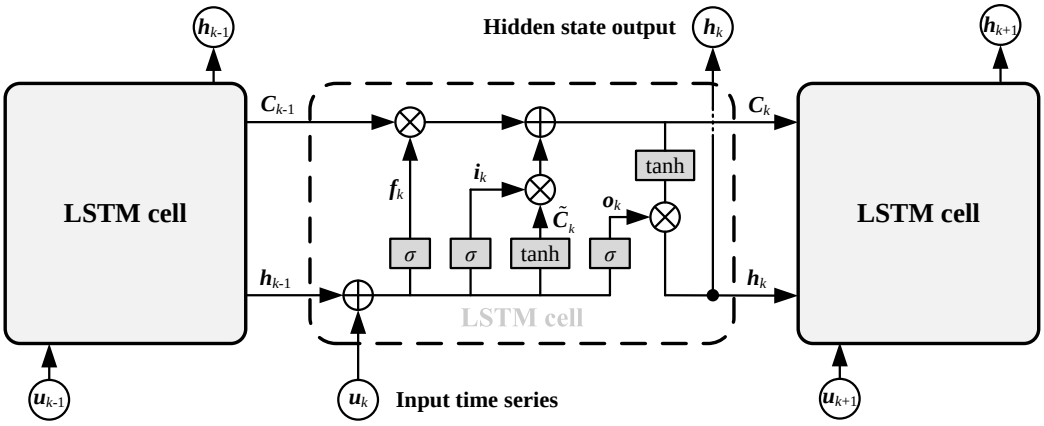


Figure 1. Structure of a standard LSTM network.

2.2.3. BiLSTM

As presented in Figure 1, the outputs of the k-th LSTM cell (i.e., $\boldsymbol{h}_k$ and $\boldsymbol{C}_k$) depend on all the inputs up to the current epoch t_k. However, future input information coming up later than the current epoch t_k can also be useful for the current output prediction [62]. To make use of all available input information, including previous, current, and future inputs, the BiLSTM network is proposed, which uses two separate LSTM networks (one for each time direction) and then merges the results [56,63]. The structure of a standard BiLSTM network is shown in Figure 2. The BiLSTM network contains two independent LSTM layers, i.e., the forward and reverse LSTM layers, with each BilSTM cell consisting of two LSTM cells, i.e., a forward LSTM cell and a reverse LSTM cell. The input and output vectors of different LSTM cells in the BiLSTM network are detailed in Table 1.

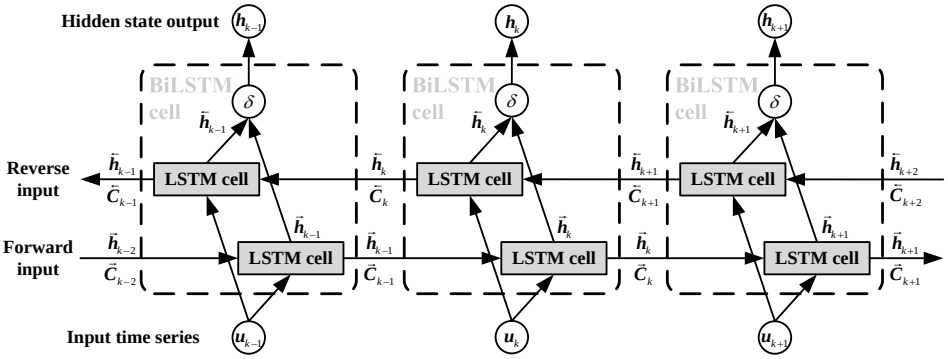


Figure 2. Structure of a standard BiLSTM network.

Table 1. Inputs and outputs of LSTM cells in the BiLSTM network.

LSRM Cell	Input Vectors	Output Vectors
k-th forward LSTM cell	current input state $\boldsymbol{u}_k$ previous hidden state $\overrightarrow{\boldsymbol{h}}_{k-1}$ previous cell state $\overrightarrow{\boldsymbol{C}}_{k-1}$	current forward hidden state $\overrightarrow{\boldsymbol{h}}_k$ current forward cell state $\overrightarrow{\boldsymbol{C}}_k$
k-th reverse LSTM cell	current input state $\boldsymbol{u}_k$ future hidden state $\overleftarrow{\boldsymbol{h}}_{k+1}$ future cell state $\overleftarrow{\boldsymbol{C}}_{k+1}$	current reverse hidden state $\overleftarrow{\boldsymbol{h}}_k$ current reverse cell state $\overleftarrow{\boldsymbol{C}}_k$

All the hidden states from the LSTM layer are arranged in chronological order and in reverse order. Then, the forward hidden state sequence $\left\{\overrightarrow{\boldsymbol{h}}_1, \cdots, \overrightarrow{\boldsymbol{h}}_k, \cdots, \overrightarrow{\boldsymbol{h}}_n\right\}$ and its reverse sequence $\left\{\overleftarrow{\boldsymbol{h}}_1, \cdots, \overleftarrow{\boldsymbol{h}}_k, \cdots, \overleftarrow{\boldsymbol{h}}_n\right\}$ are obtained. The BiLSTM hidden state $\boldsymbol{h}_k$ is computed using a connecting function $\delta(\cdot,\cdot)$, given by

$$\boldsymbol{h}_k = \delta\left(\overrightarrow{\boldsymbol{h}}_k, \overleftarrow{\boldsymbol{h}}_k\right). \tag{12}$$

3. Methodology

In this section, the methodology for improving OD accuracy using BiLSTM is provided. The proposed method improves OD accuracy by detecting inaccurate measurements and then compensating for those measurements. A modified BiLSTM neutral network is designed for inaccurate measurement detection. In this paper, the OD of a spacecraft near the Earth–Moon L2 point using angle-only measurements from a ground-based station is taken as an example to show the sample construction and training processes of the modified BiLSTM neutral network.

3.1. Dynamics and Measurement Model

The high-fidelity dynamics model of a spacecraft in cislunar includes the point-mass gravity field from the Earth, Moon, and Sun (using the ephemeris VSOP2000 [64–66]), given as

$$\begin{cases} \dot{\boldsymbol{r}}(t) = \boldsymbol{v}(t) \\ \dot{\boldsymbol{v}}(t) = -\dfrac{\mu_e}{\|\boldsymbol{r}(t)\|^3}\boldsymbol{r}(t) + \mu_m\left[\dfrac{\boldsymbol{r}_m(t)-\boldsymbol{r}(t)}{\|\boldsymbol{r}_m(t)-\boldsymbol{r}(t)\|^3} - \dfrac{\boldsymbol{r}_m(t)}{\|\boldsymbol{r}_m(t)\|^3}\right], \\ \qquad\quad + \mu_s\left[\dfrac{\boldsymbol{r}_s(t)-\boldsymbol{r}(t)}{\|\boldsymbol{r}_s(t)-\boldsymbol{r}(t)\|^3} - \dfrac{\boldsymbol{r}_s(t)}{\|\boldsymbol{r}_s(t)\|^3}\right] \end{cases} \tag{13}$$

where $\boldsymbol{r} \in \mathbb{R}^3$ and $\boldsymbol{v} \in \mathbb{R}^3$ are the position and velocity vector in Earth-centered equatorial inertial (ECI) coordinates, respectively; μ_e, μ_s and μ_m are gravitational constants of the

Earth, the Sun, and the Moon, respectively; and $\boldsymbol{r}_s(t)$ and $\boldsymbol{r}_m(t)$ label the positions of the Sun and Moon in ECI coordinates respectively.

Assume that a ground-based station can measure the line-of-sight (LOS) direction from the station to the spacecraft. The LOS direction can be represented by two angles, i.e., the right ascension (RA) α ($\alpha \in [-\pi, \pi]$) and the declination (DEC) β ($\beta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$) [67–69]. Therefore, the measurement model used in this work can be written as

$$\boldsymbol{z}(t) = \boldsymbol{h}[\boldsymbol{x}(t)] = \begin{bmatrix} \alpha(t) \\ \beta(t) \end{bmatrix} = \begin{bmatrix} \tan^{-1} \frac{y(t)-y_S(t)}{x(t)-x_S(t)} \\ \sin^{-1} \frac{z(t)-z_S(t)}{\|\boldsymbol{r}(t)-\boldsymbol{r}_S(t)\|} \end{bmatrix}, \tag{14}$$

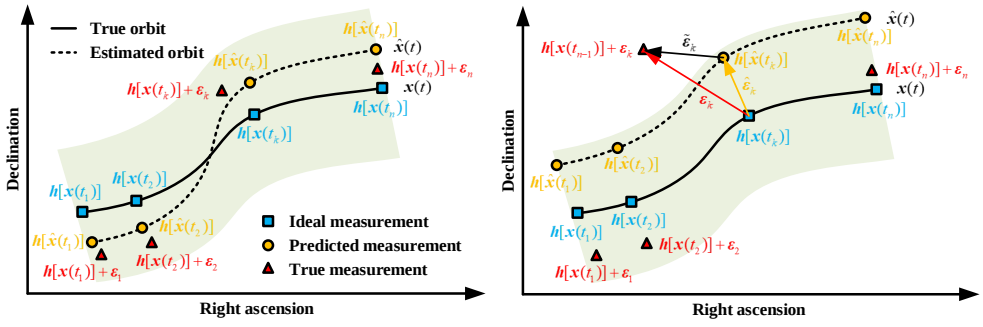
where $\boldsymbol{r}_S(t) = [x_S(t), y_S(t), z_S(t)]^T$ denotes the position vector of the station in the ECI coordinate at a specified epoch t. The standard deviation (STD) of the noises of RA and DEC are set to be $2''$, i.e., $\sigma_{\mathrm{RA}} = \sigma_{\mathrm{DEC}} = 2''$. The parameters of the ground-based station used in this work are shown in Table 2. The azimuth range restriction of the ground-based station is ignored, and only the elevation restriction is considered. The spacecraft is invisible when it is on the opposite side of Earth. Thus, the elevation should be constrained in $[0, 90]$ deg.

Table 2. Parameters of the ground-based station used.

Parameters	Value
Station	Eglin, FL
Latitude (deg)	30.57
Longitude (deg)	−86.21
Altitude (m)	34.7
Feasible elevation (deg)	0–90

3.2. Sample Construction

Careful design of the structure of the training sample is vital for the network to accurately detect inaccurate measurements. The construction of the sample dataset for the BiLSTM-based approach is illustrated in Figure 3. Figure 3a,b present the cases without and with inaccurate measurements, respectively. In Figure 3, the black solid and dashed curves denote the true orbit $\boldsymbol{x}(t)$ and the estimated orbit $\hat{\boldsymbol{x}}(t)$ obtained from the LSM, respectively.



(**a**) (**b**)

Figure 3. Illustration of the sample dataset for the BiLSTM-based approach to improve OD accuracy. (**a**) Case without inaccurate measurements; (**b**) case with inaccurate measurements.

As annotated in Figure 3, three types of measurements, the ideal measurements $\tilde{\boldsymbol{z}}_k = \boldsymbol{h}[\boldsymbol{x}(t_k)] = \left[\tilde{\alpha}_k, \tilde{\beta}_k\right]^T$, the true measurements $\boldsymbol{z}_k = \boldsymbol{h}[\boldsymbol{x}(t_k)] + \boldsymbol{\varepsilon}_k = \left[\alpha_k, \beta_k\right]^T$ and the predicted measurements $\hat{\boldsymbol{z}}_k = \boldsymbol{h}[\hat{\boldsymbol{x}}(t_k)] = \left[\hat{\alpha}_k, \hat{\beta}_k\right]^T$, are represented by blue squares, red triangles and orange circles, respectively. The light green regions are admissible regions of measurements. In this work, the admissible regions are defined as the regions where the measurement noises are smaller than two times the STD. The measurements inside and outside the admissible regions are defined as accurate and inaccurate measurements,

respectively. As shown in Figure 3a, the estimated orbit is close to the true orbit when the true measurements $\boldsymbol{z}_k$ are located in the admissible region, while the estimated errors are large when some of the true measurements are outside the admissible region (see Figure 3b).

According to Figure 3, three types of measurement errors are defined at a specified epoch t_k:

- True measurement error $\boldsymbol{\varepsilon}_k = \boldsymbol{z}_k - \tilde{\boldsymbol{z}}_k = \left[\alpha_k - \tilde{\alpha}_k, \beta_k - \tilde{\beta}_k\right]^T$. Note that the true measurement error is equivalent to the measurement noise defined in Equation (1).
- Predicted measurement error $\hat{\boldsymbol{\varepsilon}}_k = \hat{\boldsymbol{z}}_k - \tilde{\boldsymbol{z}}_k = \left[\hat{\alpha}_k - \tilde{\alpha}_k, \hat{\beta}_k - \tilde{\beta}_k\right]^T$.
- Relative measurement error $\tilde{\boldsymbol{\varepsilon}}_k = \boldsymbol{z}_k - \hat{\boldsymbol{z}}_k = \left[\alpha_k - \hat{\alpha}_k, \beta_k - \hat{\beta}_k\right]^T$.

The three types of measurement errors can be related using the following equation:

$$\tilde{\boldsymbol{\varepsilon}}_k = \boldsymbol{\varepsilon}_k - \hat{\boldsymbol{\varepsilon}}_k \,. \tag{15}$$

Among the three types of measurement errors mentioned above, the true measurement error $\boldsymbol{\varepsilon}_k$ can directly evaluate whether the true measurement $\boldsymbol{z}_k$ is inside the admissible regions. Unfortunately, the true measurement error $\boldsymbol{\varepsilon}_k$ is unavailable, as one can never know the true state of a spacecraft. Only the relative measurement error $\tilde{\boldsymbol{\varepsilon}}_k$ is available during the OD process. The relative measurement error $\tilde{\boldsymbol{\varepsilon}}_k$ is related to the true measurement error $\boldsymbol{\varepsilon}_k$, which is expected to carry the necessary information for detecting inaccurate measurements.

Next, the structure of the training sample is constructed. Apart from the relative measurement error $\tilde{\boldsymbol{\varepsilon}}_k$, there are also some variables, such as the true measurements and predicted state, that are available and might contain useful information for inaccurate measurement detection. Taking a specified epoch t_k as an instance, the learning variables of the sample dataset are given as follows:

- The relative measurement error $\tilde{\boldsymbol{\varepsilon}}_k \in \mathbb{R}^2$. The relative measurement error is assumed to contain information related to the true measurement error $\boldsymbol{\varepsilon}_k$.
- The variables related to the measurements, including the true measurements $\boldsymbol{z}_k \in \mathbb{R}^2$ and the position of the ground-based station (in the ECI coordinate) $\boldsymbol{r}_S(t_k) \in \mathbb{R}^3$. These variables may reflect the sensitivity of the estimated errors to measurement errors.
- The estimated state $\hat{\boldsymbol{x}} \in \mathbb{R}^6$ obtained using the LSM. The estimated state can reflect the physical information of the orbit, for example, the orientation and the distance of the spacecraft.
- The measurement interval $\Delta t_k = t_k - t_{k-1} \in \mathbb{R}^1$ $(k = 2, \ldots, n)$. When $n = 1$, $\Delta t_k = 0$ in this work. The measurement interval is an important variable, as the longer the measurement interval is, the weaker the relationship between two successive measurements.

With the aforementioned learning variables, the input vector of the sample dataset at a specified epoch t_k is written as

$$\boldsymbol{s}_k = [\tilde{\boldsymbol{\varepsilon}}_k; \boldsymbol{z}_k; \boldsymbol{r}_S(t_k); \hat{\boldsymbol{x}}(t_k); \Delta t_k] \in \mathbb{R}^{14} \,. \tag{16}$$

Inaccurate measurement detection is essentially a binary classification problem. Motivated by [70], the output vector of the sample dataset at epoch t_K is expressed as

$$\boldsymbol{y}_k = \left[p_k^{\mathrm{RA}}, q_k^{\mathrm{RA}}, p_k^{\mathrm{DEC}}, q_k^{\mathrm{DEC}}\right]^T \in \mathbb{R}^4 \,, \tag{17}$$

where $p_k^{\mathrm{RA}} \in \{0,1\}$, $q_k^{\mathrm{RA}} \in \{0,1\}$, $p_k^{\mathrm{DEC}} \in \{0,1\}$ and $q_k^{\mathrm{DEC}} \in \{0,1\}$. $p_k^{\mathrm{RA}} \geq q_k^{\mathrm{RA}}$ and $p_k^{\mathrm{RA}} > q_k^{\mathrm{RA}}$ mean that RA α_k is inside and outside the admissible region, respectively, whereas $p_k^{\mathrm{DEC}} \geq q_k^{\mathrm{DEC}}$ and $p_k^{\mathrm{DEC}} > q_k^{\mathrm{DEC}}$ mean that DEC β_k is inside and outside the admissible region, respectively. This type of output form is particularly common in binary classification problems. In this way, the cross entropy can be used as a loss function to train the network.

Note that the output $\boldsymbol{y}_k$ at the current epoch t_k is related not only to the current input $\boldsymbol{s}_k$, but also to the inputs at the previous and future epochs. By collecting all the inputs $\boldsymbol{s}_k$ $(k = 1, \cdots, n)$ and sequentially processing them, the output $\boldsymbol{y}_k$ can be better predicted. Thus, the structure of the sample dataset is written as

$$\chi = \left\{ \underbrace{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_n}_{S} \right\} \mapsto \left\{ \underbrace{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n}_{Y} \right\}, \tag{18}$$

where χ denotes the sample dataset, and $S = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n\}$ and $Y = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\}$ are the input and output sets of the sample dataset χ, respectively. Note that the number of inputs and outputs in one sample χ, n, is not fixed in our work, as the number of incoming measurements in an OD process is arbitrary.

Based on the above discussion on sample input and output, a method is proposed for effectively generating a large number of samples herein. First, three groups of Earth–Moon LPOs, i.e., a group of Southern L2 Halo orbits, a group of Northern L2 Halo orbits, and a group of L2 Lyapunov orbits, are generated [71]. The nondimensional orbits in the circle restrict three-body problem (CRTBP) are shown in Figure 4. The corresponding nondimensional initial conditions, periods, and Jacobi constants of the Southern L2 Halo and L2 Lyapunov orbits are listed in Table 3 and Table 4, respectively, where T and J denote the orbit period and Jacobi constant, respectively. The Northern L2 Halo family can be obtained by reflecting the southern Halo family across the x-y plane.

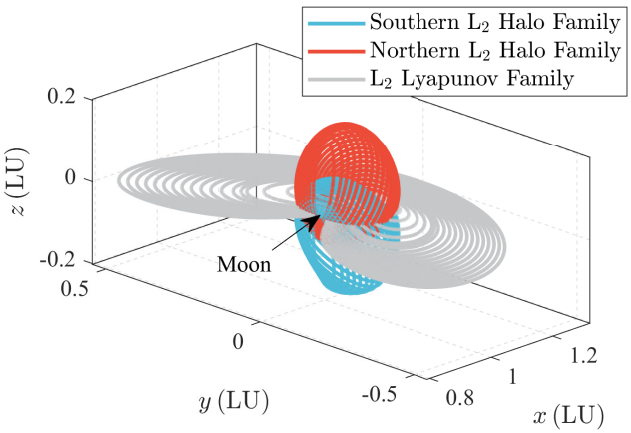


Figure 4. An illustration of orbit families for sample generation.

Table 3. Orbit parameters of the Southern L2 Halo family [71,72].

Orbit	x_0	z_0	$\dot{y}_0$	T	J
1	1.1809	0	−0.1558	3.4155	3.1641
2	1.1807	−0.0139	−0.1569	3.4139	3.1632
3	1.1802	−0.0259	−0.1596	3.4100	3.1611
4	1.1794	−0.0379	−0.1636	3.4037	3.1579
5	1.1782	−0.0499	−0.1687	3.3949	3.1534
6	1.1766	−0.0619	−0.1747	3.3835	3.1479
7	1.1746	−0.0739	−0.1811	3.3694	3.1414
8	1.1721	−0.0859	−0.1878	3.3521	3.1340
9	1.1690	−0.0979	−0.1945	3.3314	3.1257
10	1.1653	−0.1099	−0.2010	3.3066	3.1166

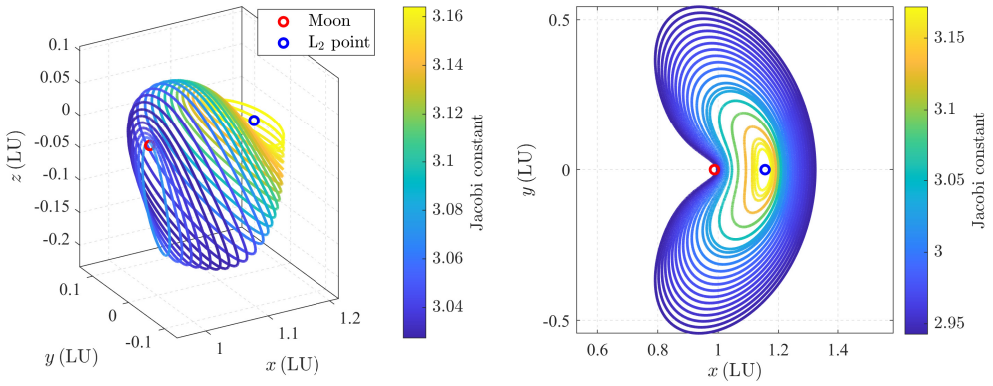
Table 3. *Cont.*

Orbit	x_0	z_0	$\dot{y}_0$	T	J
11	1.1611	−0.1219	−0.2072	3.2768	3.1068
12	1.1561	−0.1339	−0.2129	3.2406	3.0963
13	1.1503	−0.1459	−0.2179	3.1960	3.0853
14	1.1435	−0.1579	−0.2219	3.1395	3.0737
15	1.1354	−0.1699	−0.2247	3.0647	3.0617
16	1.1234	−0.1837	−0.2252	2.9380	3.0472
17	1.1113	−0.1934	−0.2220	2.7897	3.0367
18	1.0992	−0.1993	−0.2152	2.6245	3.0302
19	1.0872	−0.2020	−0.2053	2.4486	3.0273
20	1.0759	−0.2022	−0.1932	2.2778	3.0276
21	1.0633	−0.2003	−0.1769	2.0876	3.0308
22	1.0512	−0.1968	−0.1586	1.9106	3.0363
23	1.0392	−0.1919	−0.1379	1.7420	3.0439
24	1.0274	−0.1856	−0.1147	1.5827	3.0533
25	1.0118	−0.1739	−0.0798	1.3741	3.0700

Table 4. Orbit parameters of the Lyapunov family [71,72].

Orbit	x_0	$\dot{y}_0$	T	J
1	1.1762	−0.1228	3.3979	3.1718
2	1.1809	−0.1558	3.4155	3.1641
3	1.1843	−0.1815	3.4339	3.1565
4	1.1924	−0.2498	3.5122	3.1302
5	1.2005	−0.3235	3.6831	3.0918
6	1.2086	−0.3801	3.9409	3.0565
7	1.2167	−0.4162	4.2071	3.0329
8	1.2200	−0.4275	4.3112	3.0257
9	1.2248	−0.4418	4.4573	3.0166
10	1.2329	−0.4620	4.6912	3.0045
11	1.2410	−0.4792	4.9113	2.9948
12	1.2491	−0.4947	5.1195	2.9869
13	1.2572	−0.5090	5.3172	2.9799
14	1.2653	−0.5226	5.5053	2.9739
15	1.2734	−0.5356	5.6848	2.9685
16	1.2815	−0.5482	5.8560	2.9634
17	1.2896	−0.5606	6.0196	2.9589
18	1.2977	−0.5728	6.1758	2.9545
19	1.3058	−0.5848	6.3251	2.9505
20	1.3139	−0.5967	6.4678	2.9466
21	1.3220	−0.6086	6.6042	2.9427

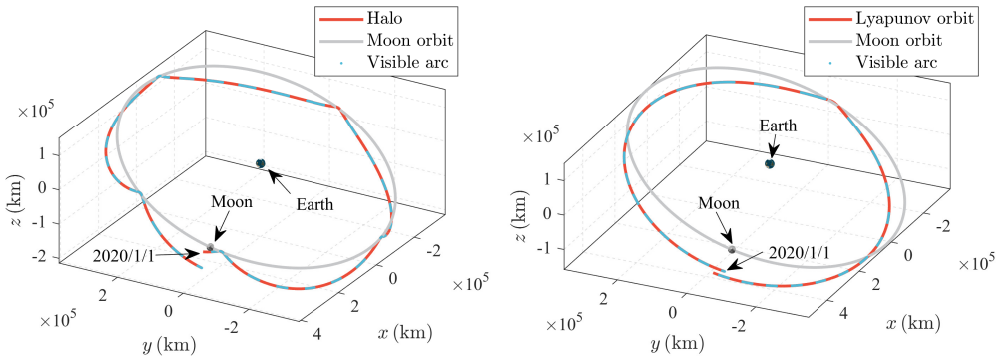
Moreover, the nondimensional Southern L2 Halo and L2 Lyapunov orbits in the Earth–Moon CRTBP are plotted in Figure 5a and Figure 5b, respectively, with the colors representing the Jacobi constant of the orbits. According to Tables 3 and 4, 25 Southern L2 Halo orbits, 25 Northern L2 Halo orbits, and 21 L2 Lyapunov orbits are obtained. Notice that the second L2 Lyapunov orbit repeatedly appears in the southern and Northern L2 Halo families. Therefore, 69 nondimensional periodic orbits are available after the repetitive orbits are removed.



(**a**) (**b**)

Figure 5. Earth–Moon Libration point orbits in the nondimensional rotating coordinate. (**a**) Earth–Moon L2 Halo orbits; (**b**) Earth–Moon L2 Lyapunov orbits.

Second, these nondimensional periodic orbits are corrected in the high-fidelity ephemeris force model (i.e., Equation (13)) using the two-level differential corrector [73], after which 69 ephemeris trajectories are computed. These ephemeris trajectories are propagated using the Runge–Kutta (4,5) solver [74] with a relative tolerance of 10^{-12} and an absolute tolerance of 10^{-12}, with an initial epoch of 1 Jan 2020 and a propagation period of 1 month (approximately 27.32 days). The ephemeris trajectories of the 25th Halo orbit and 21st Lyapunov orbit are presented in Figure 6a and Figure 6b, respectively.



(**a**) (**b**)

Figure 6. Earth–Moon Libration point orbits in the ECI coordinate. (**a**) The 25th Earth–Moon L2 Halo orbit; (**b**) The 21st Earth–Moon L2 Lyapunov orbit.

Next, each ephemeris trajectory is discretized into several points, with a constant interval of 6 h. In this way, 7590 points are obtained. For each point, one sample is constructed using the following steps:

Step 1: The state and associated epoch of the point are taken as the initial state and epoch of one OD, i.e., $\boldsymbol{x}_0$ and t_0. The time of flight (i.e., the measurement duration) Tof, the number of measurements n, and the initial estimated error $\boldsymbol{e}_0 = [\boldsymbol{e}_{r,0}; \boldsymbol{e}_{v,0}] \in \mathbb{R}^6$ are

randomly generated based on the lower and upper bounds listed in Table 5. The final epoch is calculated as $t_f = t_0 + Tof$. The first and final measurements are assumed to be available at the initial and epochs, i.e., $t_1 = t_0$ and $t_n = t_f$. The epochs of the remaining $n-2$ measurements, t_k ($k = 2, \cdots, n-1$) are randomly generated using the Latin hypercube sampling method [75] over the duration (t_1, t_n).

Table 5. Parameter ranges for generating samples.

Parameters	Symbol	Lower Bounds	Upper Bounds
Time of flight (h)	$Tof \in \mathbb{R}^1$	2	6
Number of measurements (-)	$n \in \mathbb{R}^1$	121	361
Initial position estimated error (km)	$\boldsymbol{e}_{r,0} \in \mathbb{R}^3$	$[-300, -300, -300]$	$[300, 300, 300]$
Initial velocity estimated error (m/s)	$\boldsymbol{e}_{v,0} \in \mathbb{R}^3$	$[-30, -30, -30]$	$[30, 30, 30]$

Step 2: Propagate the true orbit $\boldsymbol{x}_0$ using the high-fidelity ephemeris force model in the duration $[t_1, t_n]$. Check the visibility over the duration $[t_1, t_n]$ according to the feasible elevation range in Table 2. If visible, go to Step 3.

Step 3: Calculate the ideal measurement $\tilde{\boldsymbol{z}}_k = \boldsymbol{h}[\boldsymbol{x}(t_k)]$ at n epochs t_k ($k = 1, \cdots, n$). Randomly generate the true measurement error $\boldsymbol{\varepsilon}_k$ according to the given STDs (i.e., $\sigma_{\mathrm{RA}} = \sigma_{\mathrm{DEC}} = 2''$). Compute the true measurement $\boldsymbol{z}_k = \boldsymbol{h}[\boldsymbol{x}(t_k)] + \boldsymbol{\varepsilon}_k$. Check whether the true measurement error $\boldsymbol{\varepsilon}_k$ exceeds the admissible region. If the true measurement error of RA is inside the admissible region, $p_k^{\mathrm{RA}} = 1$ and $q_k^{\mathrm{RA}} = 0$, whereas if the true measurement error of RA is outside the admissible region, $p_k^{\mathrm{RA}} = 0$ and $q_k^{\mathrm{RA}} = 1$, which are the same for DEC. Then, the output set $S = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n\}$ is obtained.

Step 4: The initial estimated state is obtained using $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0 + \boldsymbol{e}_0$. Compute the optimal estimated state $\hat{\boldsymbol{x}}(t)$ using the LSM. The input set $Y = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\}$ is then constructed using Equations (16) and (17).

Step 5: A sample is constructed as $\chi = \{S \mapsto Y\}$.

Using the above process, 3761 valid samples are obtained. All these samples, including the inputs and outputs, are normalized to $[0, 1]$. Notice that in the ECI coordinate $z_S(t)$ is a constant. Thus, in this work, $z_S(t)$ is normalized to 0.5. In the following subsections, the modified BiLSTM network is trained using these samples.

3.3. Modified BiLSTM Structure

With the sample χ from Equation (18) given, the structure of the designed modified BiLSTM network is displayed in Figure 7. Apart from n BiLSTM cells, the modified BiLSTM network also contains two initialization layers, a pre-BiLSTM layer, a post-BiLSTM layer, and a sigmoid layer.

As shown in Figure 7, the pre-BiLSTM layer is applied to the first layer of the designed network, which is employed to extract the useful features from the sequential input $\{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n\}$ (as defined in Equation (16)). The BiLSTM cells serve as the second layer, which is expected to fully learn information extracted by the pre-BiLSTM layer. The forward and backward association characteristics of the relative measurement errors, true measurements, station positions, estimated states, and measurement intervals of different epochs are abstracted and merged by the BiLSTM cells. Then, using a post-BilSTM layer, the hidden states obtained from the BiLSTM cells are reweighted. In this way, the key features will be given more weight to reduce interferences from inconsequential information and improve the classification performance. The sigmoid layer is the last layer in the designed network, which is a typical operation in binary classifications. The sigmoid function is a mathematical function that takes any real number and "squashes" it into a range between 0 and 1. It is often used in machine learning and neural networks to introduce non-linearity and convert an input into a probability [51,76]. The function has an S-shaped curve and is expressed as $f(x) = 1/(1 + e^{(-x)})$. This function is particularly useful for scenarios where we need to make predictions or classify data into two categories. Its smooth gradient also makes it useful for efficient learning during backpropagation in neural networks. In

addition, two initialization layers are added to the network to generate the hidden states $\overrightarrow{h}_0$ and $\overleftarrow{h}_{n+1}$. The input vectors of two initialization layers, s_0 and s_{n+1}, are assumed to be 14-dimensional zero vectors, i.e., $s_0 = s_{n+1} = \mathbf{0}_{14\times 1} \in \mathbb{R}^{14}$.

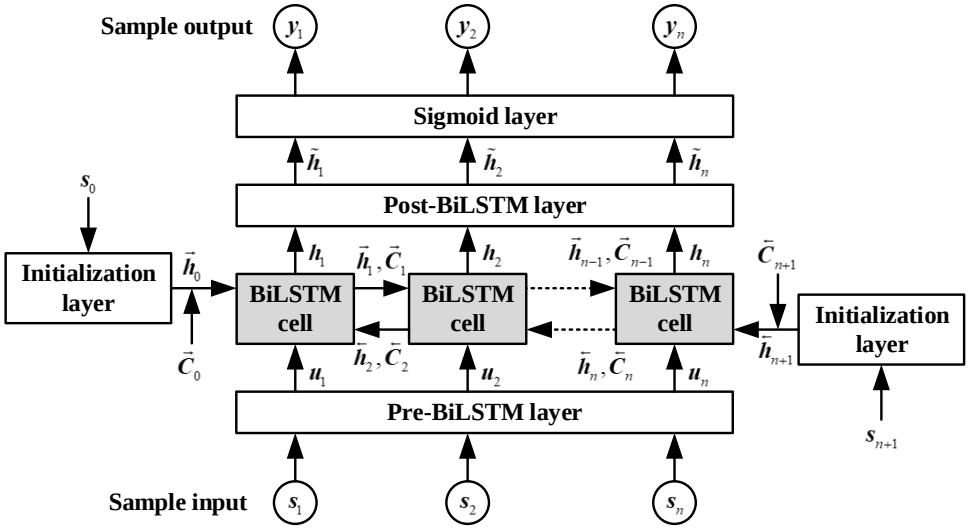


Figure 7. Structure of the designed modified BiLSTM network.

The initialization, pre-BiLSTM, post-BiLSTM, and sigmoid layers are designed as deep neural networks (DNNs). A standard DNN contains input, output, and hidden layers [42], with its structure presented in Figure 8. In Figure 8, each circle represents a neuron. The sigmoid layer is designed as a three-layer fully connected DNN with the sigmoid as the activation function. The initialization, pre-BiLSTM and post-BiLSTM layers are assumed to have the same structure in their hidden layers, i.e., the number of hidden layers, the number of neurons and the activation function used are the same. Let L and M denote the number of hidden layers and the corresponding number of neurons in one hidden layer, respectively. Moreover, the dimensions of the hidden and cell states of the BiLSTM cells are also set to M, and $\overrightarrow{C}_0$ and $\overleftarrow{C}_{n+1}$ are assumed to be M-dimensional zero vectors. According to the above discussion, the structures of the four layers are shown in Table 6. Notice that in this work, two independent networks are trained for anomaly detection of RA and DEC; therefore, the dimension of the predictions of the proposed modified BiLSTM network is 2.

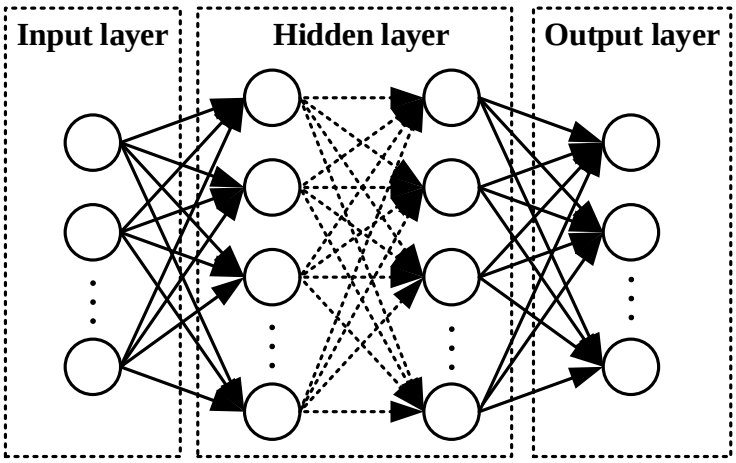


Figure 8. Structure of a standard DNN network.

Table 6. Structures of the initialization, pre-BiLSTM, post-BiLSTM, and sigmoid layers.

	Number of Neurons of Input Layer	Number of Hidden Layers	Number of Neurons of Hidden Layer	Number of Neurons of Output Layer	Mapping Relationship
Initialization lager	14	L	M	M	$\boldsymbol{s}_0 \mapsto \overrightarrow{\boldsymbol{h}}_0 : \mathbb{R}^{14} \mapsto \mathbb{R}^M$
	14	L	M	M	$\boldsymbol{s}_{n+1} \mapsto \overleftarrow{\boldsymbol{h}}_{n+1} : \mathbb{R}^{14} \mapsto \mathbb{R}^M$
Pre-BiLSTM layer	14	L	M	M	$\boldsymbol{s}_k \mapsto \boldsymbol{u}_k : \mathbb{R}^{14} \mapsto \mathbb{R}^M$
Post-BiLSTM layer	M	L	M	M	$\boldsymbol{h}_k \mapsto \tilde{\boldsymbol{h}}_k : \mathbb{R}^{14} \mapsto \mathbb{R}^M$
Sigmoid layer	M	1	M	2	$\boldsymbol{h}_k \mapsto \boldsymbol{y}_k : \mathbb{R}^M \mapsto \mathbb{R}^2$

3.4. Training Process and Results

Before training the modified BiLSTM network, the 3761 samples are randomly divided into two sample datasets: one training dataset and one validation dataset. The training dataset contains 80% of the samples (3009 samples), and the validation dataset has the remaining 20% (662 samples). The training dataset is used to train the modified BiLSTM network, whereas the validation dataset is used to test its accuracy. In this paper, the modified BiLSTM networks for inaccurate RA and DEC detection are called RA BiLSTM and DEC BiLSTM, respectively. In this work, the construction and training of the modified BiLSTM network are based on the Python implementation of PyTorch. The cross-entropy loss is employed as the loss function during the training process. For RA, the cross-entropy loss of one sample is written as

$$\text{Cross-entropy loss} = \sum_{k=1}^{n} \left(q_k^{\text{RA}} \ln \hat{p}_k^{\text{RA}} + p_k^{\text{RA}} \ln \hat{q}_k^{\text{RA}} \right), \tag{19}$$

whereas for DEC, the cross-entropy loss of one sample is given as

$$\text{Cross-entropy loss} = \sum_{k=1}^{n} \left(q_k^{\text{DEC}} \ln \hat{p}_k^{\text{DEC}} + p_k^{\text{DEC}} \ln \hat{q}_k^{\text{DEC}} \right), \tag{20}$$

where $\{\hat{p}_k^{\text{RA}}, \hat{q}_k^{\text{RA}}\}$ and $\{\hat{p}_k^{\text{DEC}}, \hat{q}_k^{\text{DEC}}\}$ are the predictions of the RA BiLSTM and DEC BiLSTM, respectively.

In addition, the maximal iteration is set to 100. In each iteration, the 3009 samples from the training dataset are sequentially trained. The parameters of the modified BiLSTM networks are updated once one sample is trained. Thus, the parameters are updated 3009 times in one iteration. The learning rate is initialized as 10^{-4} and decayed by 0.98 every iteration, given as

$$\text{lr} = 0.98^{\text{iter}-1} \times 10^{-4}, \tag{21}$$

where lr and iter denote the learning rate and the number of iterations, respectively.

In each iteration, the cross-entropy loss of 3009 training samples and 662 validation samples are recorded. The mean cross-entropy (MCE) losses of the training and validation datasets are then calculated. The network parameters with minimal validation MCE are retained and used in the subsequent inaccurate measurement detection. It takes approximately 2 h to train the neural network using a workstation (CPU: 13th Gen Intel(R) Core(TM) i9-13900KF; GPU: NVIDIA GeForce RTX 4090). The training results of the modified BiLSTM networks with different structures are listed in Table 7. Several conclusions can be drawn from Table 7. First, for the concerned anomaly detection problem, the network with one hidden layer performs better than that with two hidden layers. Second, the minimal validation MCEs using Leaky ReLU are smaller than those using ReLU and Tanh. Third, the networks with 32 neurons per layer are more accurate than those with 16, 64, and 128 neurons per layer. Therefore, the modified BiLSTM network with one hidden layer,

thirty-two neurons in the hidden layer, and the Leaky ReLU activation function is chosen to detect inaccurate measurements in this work.

Table 7. Training results of networks with different structures.

L	*M*	Activation Function	Minimal Validation MCE	
			RA	DEC
1	16	Leaky ReLU	2.5320×10^{-2}	1.1904×10^{-1}
		ReLU	1.1746×10^{-1}	2.5989×10^{-2}
		Tanh	2.7256×10^{-2}	1.1799×10^{-1}
	32	Leaky ReLU	2.292×10^{-2}	2.5028×10^{-2}
		ReLU	2.4641×10^{-2}	3.3524×10^{-2}
		Tanh	2.4527×10^{-2}	2.6765×10^{-2}
	64	Leaky ReLU	2.3771×10^{-2}	2.7930×10^{-2}
		ReLU	2.3828×10^{-2}	2.8491×10^{-2}
		Tanh	2.8891×10^{-2}	1.1901×10^{-1}
	128	Leaky ReLU	2.6264×10^{-2}	2.3285×10^{-2}
		ReLU	2.4589×10^{-2}	2.3786×10^{-2}
		Tanh	2.4291×10^{-2}	2.3836×10^{-2}
2	16	Leaky ReLU	1.2170×10^{-1}	2.6885×10^{-2}
		ReLU	2.4529×10^{-2}	1.2117×10^{-1}
		Tanh	2.5764×10^{-2}	1.8663×10^{-1}
	32	Leaky ReLU	1.1737×10^{-1}	1.1863×10^{-1}
		ReLU	2.4806×10^{-2}	1.1755×10^{-2}
		Tanh	2.3103×10^{-2}	1.8573×10^{-2}
	64	Leaky ReLU	2.4558×10^{-2}	2.4421×10^{-2}
		ReLU	2.5516×10^{-2}	1.8231×10^{-1}
		Tanh	2.6790×10^{-2}	1.8164×10^{-1}
	128	Leaky ReLU	2.7573×10^{-2}	3.3301×10^{-2}
		ReLU	2.6885×10^{-2}	2.3767×10^{-2}
		Tanh	2.7142×10^{-2}	1.1875×10^{-1}

The time histories of the MCE of the chosen network are plotted in Figure 9. It can be seen from Figure 9 that the training and validation MCEs decrease synchronously as the iteration increases. The modified BiLSTM networks have not overfitted during the training process, meaning that the design of the inputs, outputs, and structure of the networks is reasonable.

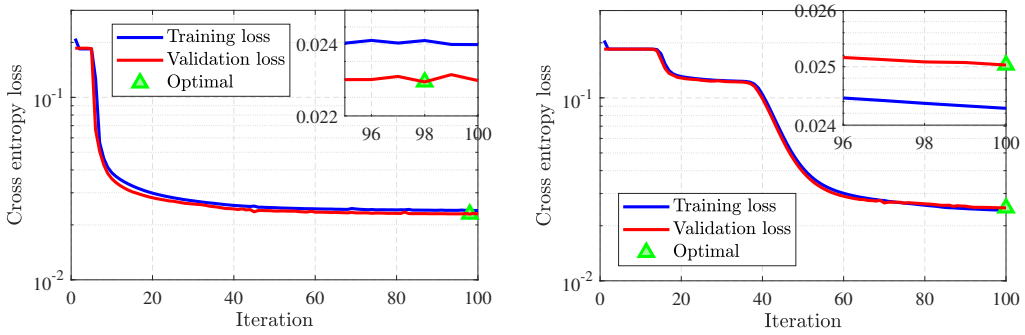


Figure 9. Training convergence history. The green triangle represents the iteration of the minimal validation MCE. (**a**) RA, (**b**) DEC.

To verify the classification accuracy of the trained network, 3629 new samples that are different from the previously obtained 3761 samples are randomly generated using

the same method in Section 3.2. The trained networks are employed to detect inaccurate measurements for each newly generated sample. The accuracy rate (AR) is employed as the metric to evaluate the performance of the networks, which is defined as the percentage of all the measurements belonging to each class that are correctly classified. Taking a specified sample with n inputs and n outputs as an example, the AR of the inaccurate class is defined as

$$\mathrm{AR} = \frac{\tilde{n}_{\text{inaccurate class}}}{n_{\text{inaccurate class}}} \times 100\%, \tag{22}$$

where $n_{\text{inaccurate class}}$ and $\tilde{n}_{\text{inaccurate class}}$ ($\tilde{n}_{\text{inaccurate class}} \leq n_{\text{inaccurate class}}$) represent the number of inaccurate measurements and the number of inaccurate measurements that are correctly classified, respectively.

The ARs of 3629 newly generated samples are computed and shown in Figure 10. Scatter plots are provided in the left subplots of Figure 10a,b, in which the x- and y-axes represent the number of measurements and the AR, respectively. Additionally, violin plots are provided in the right subplots of Figure 10a,b. The violin plot depicts distributions of the corresponding ARs using density curves. The width of each curve corresponds with the approximate frequency of the ARs in each region. In the middle of each density curve is a small box plot, with the rectangle showing the ends of the first and third quartiles and the central showing the median. It can be seen from Figure 10 that the ARs of the accurate class are higher than 95%. For most samples, the ARs of the inaccurate class are higher than 80%. In addition, the predicted accuracy is better as the number of measurements increases.

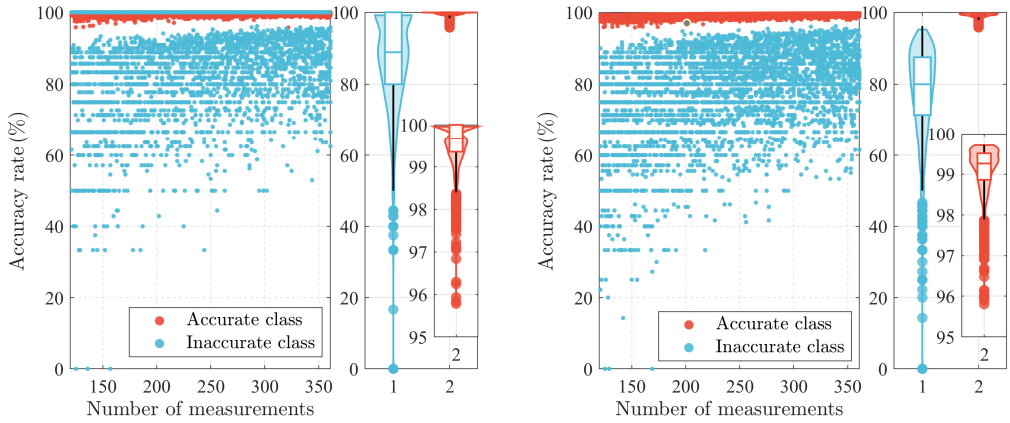


Figure 10. Accuracy test results of 3629 newly generated samples. (**a**) RA; (**b**) DEC.

Furthermore, the classification results of 905018 measurements (3629 samples, with each sample containing 121–361 measurements) are shown in Tables 8 and 9. The true positive rate (TPR) and false negative rate (FNR) represent the percentages of all the measurements belonging to each class that are correctly and incorrectly classified, respectively. The TPRs of the accurate and inaccurate classes are approximately 99.6% and 87%, respectively. The overall accuracies are higher than 99%. These results mean that it is almost impossible for the trained networks to wrongly identify an accurate measurement as inaccurate, whereas they may miss approximately 13% of the inaccurate measurements.

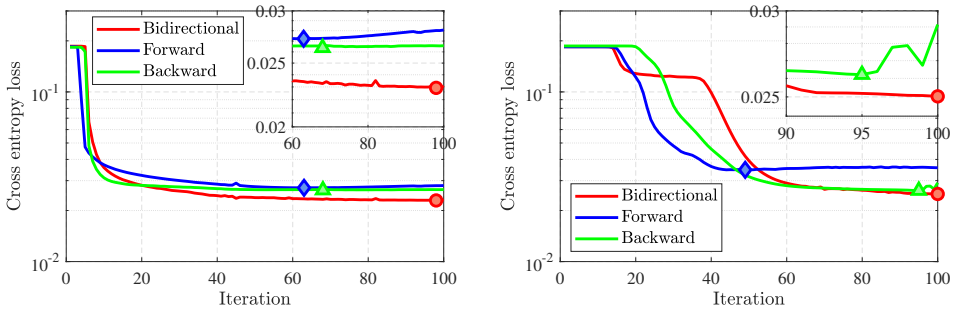
Table 8. Accuracy test results of RA BiLSTM (overall accuracy = 99.0633%).

Actual Class	Total	Predicted Class		True Positive Rate (%)	False Negative Rate (%)
		Accurate	Inaccurate		
Accurate	864,022	860,716	3306	99.6173	0.3826
Inaccurate	40,996	5171	35,825	87.3865	12.6134

Table 9. Accuracy test results of DEC BiLSTM (overall accuracy = 99.0039%).

Actual Class	Total	Predicted Class		True Positive Rate (%)	False Negative Rate (%)
		Accurate	Inaccurate		
Accurate	863,856	860,438	3418	99.6043	0.3956
Inaccurate	41,162	5596	35,566	86.4049	13.5950

Finally, two regular unidirectional LSTMs, i.e., a forward LSTM and a backward LSTM, are trained using the same training dataset, with the time histories of the validation losses and the TPRs shown in Figure 11 and Table 10, respectively. The number of hidden layers, the number of neurons, and the activation function of the forward and backward LSTMs are the same as those in the modified BiLSTM network. The validation losses and the TPRs of the modified BiLSTM network are plotted in Figure 11 and Table 10 as a comparison. Several interesting properties of the modified BiLSTM network in general can be seen from Figure 11 and Table 10. First, the BiLSTM network has a smaller validation loss and higher TPRs than the other two unidirectional LSTMs, indicating that the bidirectional structure has a better classification capacity. The second is that BiLSTM converges slower than the forward and backward LSTMs, as the bidirectional structure is more complex than the forward and backward structures. Although the BiLSTM has better training convergence and better accuracy, it requires more computational burden when training. In this work, the computational time of training a BiLSTM is approximately 2 times the computational time of training a unidirectional LSTM.



(**a**) (**b**)

Figure 11. Validation losses of the bidirectional LSTM, forward LSTM, and backward LSTM. The minimal validation MCEs of the three LSTMs are represented by the red circle, blue diamond, and green triangle, respectively. (**a**) RA; (**b**) DEC.

Table 10. True positive rate results of different network structures.

		Accurate Class	Inaccurate Class
BiLSTM	RA	99.6173	87.3865
	DEC	99.6043	86.4049
Forward LSTM	RA	99.4742	83.5989
	DEC	95.4365	34.5017
Backward LSTM	RA	99.5834	84.4987
	DEC	95.5991	64.3418

3.5. Overall Methodology

Using the trained modified BiLSTM networks, inaccurate measurements can be detected. Then, a method is proposed to compensate for these inaccurate measurements to improve OD accuracy. Take the epoch t_k as an instance. If the RA α_k is classified as inaccurate (i.e., the measurement error exceeds $2\sigma_{\mathrm{RA}}$), the following operation is implemented:

$$\alpha_k \leftarrow \alpha_k - \text{sign}\left(\tilde{\varepsilon}_k^{\text{RA}}\right) \cdot \sigma_{\text{RA}}\,, \tag{23}$$

where $\tilde{\varepsilon}_k^{\text{RA}} = \alpha_k - \hat{\alpha}_k$ represents the relative measurement error of RA and $\text{sign}(\cdot)$ denotes the sign function, expressed as

$$\text{sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}. \tag{24}$$

In addition, if DEC is detected as inaccurate, the DEC is compensated using

$$\beta_k \leftarrow \beta_k - \text{sign}\left(\tilde{\varepsilon}_k^{\text{DEC}}\right) \cdot \sigma_{\text{DEC}}\,, \tag{25}$$

where $\tilde{\varepsilon}_k^{\text{DEC}} = \beta_k - \hat{\beta}_k$ represents the relative measurement error of DEC.

The reason for using Equations (23)–(25) is that, although the relative measurement errors are not equivalent to the true ones, they should have the same signs. Therefore, the signs of the relative measurement errors are employed to provide the directions of the measurement compensation. Moreover, the magnitudes of the measurement compensation are set as the STD of the measurement error. This is because, although the overall accuracies of the trained modified BiLSTM networks are higher than 99%, a few measurements belonging to the accurate class are still incorrectly classified. These incorrectly classified measurements have true measurements smaller than $4''$ (i.e., two times the STD of measurement errors). Thus, the magnitude of one STD is used to avoid negative influences from those incorrectly classified measurements.

In this paper, the LSM using BiLSTM-based inaccurate measurement detection is denoted as BiLSTM-LSM. The overall procedure of the proposed BiLSTM-LSM for improving OD accuracy is shown in Algorithm 2. The optimal estimations obtained by the LSM with and without BiLSTM-based inaccurate measurement detection are distinguished using the superscripts 'BiLSTM-LSM' and 'LSM'.

Algorithm 2: Pseudocode of the proposed method

Input: dynamics model $\boldsymbol{f}[\boldsymbol{x}(t), t]$, measurement model $\boldsymbol{h}[\boldsymbol{x}(t)]$, n measurements $\boldsymbol{z}_k$, initial estimated state $\hat{\boldsymbol{x}}_0$, measurement STD σ_{RA} and σ_{DEC}

Output: the improved optimal estimation $\hat{\boldsymbol{x}}_0^{\text{BiLSTM-LSM}}$

1 On receipt of the n measurements $\boldsymbol{z}_k$ and the initial estimation $\hat{\boldsymbol{x}}_0$, calculate the converged optimal estimations $\hat{\boldsymbol{x}}_0^{\text{LSM}}$ and $\hat{\boldsymbol{x}}^{\text{LSM}}(t)$ using the LSM;

2 Calculate the relative measurement error $\tilde{\boldsymbol{\varepsilon}}_k$, the position of the ground-based station $\boldsymbol{r}_S(t_k)$, measurement interval Δt_k. Generate the input series $\boldsymbol{s}_k \in \mathbb{R}^{14}$;

3 Detect the inaccurate measurements using the trained modified BiLSTM network;

4 Compensate the detected inaccurate measurements using Equations (23)–(25);

5 With compensated measurements obtained, the improved optimal estimations $\hat{\boldsymbol{x}}_0^{\text{BiLSTM-LSM}}$ and $\hat{\boldsymbol{x}}^{\text{BiLSTM-LSM}}(t)$ are computed using the LSM. Notice that in this step, the converged optimal estimations $\hat{\boldsymbol{x}}_0^{\text{LSM}}$ in **step 1** can be utilized as the initial estimation to speed up the convergence.

4. Numerical Simulation

In this section, a cislunar periodic orbit case is investigated to show the advantages of the proposed BiLSTM-LSM in improving OD accuracy. The nominal orbit is a 4:1 synodic resonant NRHO. Note that the 4:1 NRHO is not used in the training or validation datasets. Thus, it is easier to show the generalization capacity of the proposed method in scenarios that have not been trained. In this work, the MATLAB (version R2022a) is employed to generate samples, carry out the estimation process, and plot results, and PyCharm (Python 3.10) is employed to construct and train the neural networks.

4.1. Scenario Descriptions

The nominal 4:1 synodic resonant NRHOs are shown in Figure 12, where the solid red line and blue dots represent the nominal obit and the visible arcs, respectively. The beginning epoch of the OD simulation is selected from a perilune of the 4:1 synodic resonant NRHO. The initial epoch and the associated initial state in the ECI coordinate are listed in Table 11. The measurement duration and the number of measurements are set as 6 h and 361, respectively.

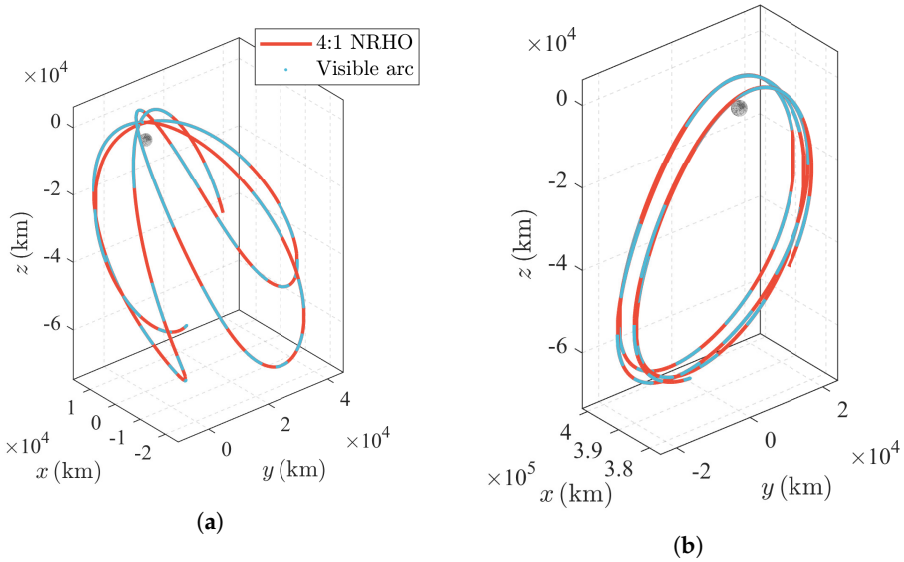


Figure 12. The 4:1 synodic resonant NRHO. (**a**) Moon-centered inertial frame; (**b**) Earth–Moon rotating frame.

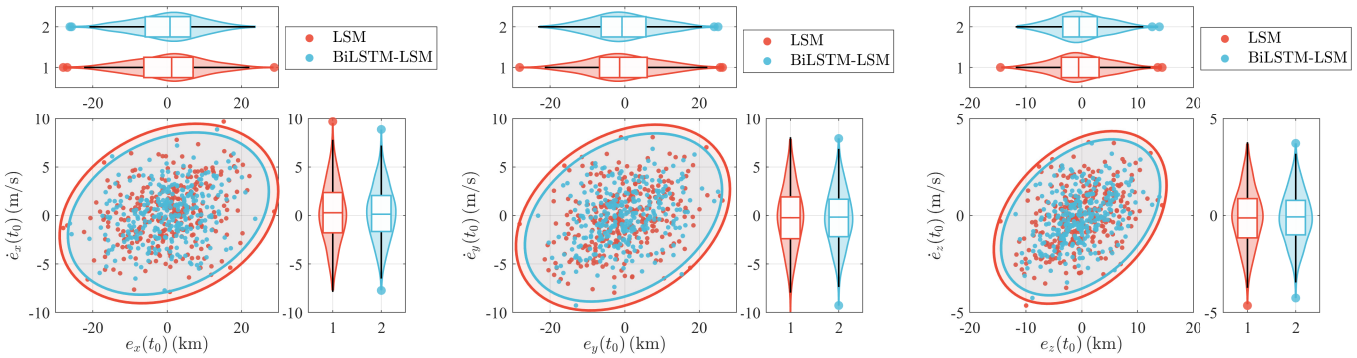
Table 11. Parameters of the simulated case.

Parameter		Value
Initial epoch		2,458,860.75 Julian Date
Time of flight		6 h
Number of measurements		361
Initial state (ECI frame)	x_0	−238,078.6112 km
	y_0	251,708.0350 km
	z_0	132,135.5595 km
	$\dot{x}_0$	−1.5244 km/s
	$\dot{y}_0$	−0.8960 km/s
	$\dot{z}_0$	−0.8935 km/s

4.2. Simulation Results

Three hundred Monte Carlo (MC) runs are performed using both LSM and the proposed BiLSTM-LSM methods. For each MC run, the intervals between two successive measurements and the true measurement errors are randomly generated. In addition, the initial estimated errors are generated using the bounds listed in Table 5. The MC simulation results of the conventional LSM and the proposed BiLSTM-LSM methods are summarized in Figure 13. In Figure 13, $\left[e_x(t_0), e_y(t_0), e_z(t_0)\right]^T = \hat{\boldsymbol{r}}_0 - \boldsymbol{r}_0$ and $\left[\dot{e}_x(t_0), \dot{e}_y(t_0), \dot{e}_z(t_0)\right]^T = \hat{\boldsymbol{v}}_0 - \boldsymbol{v}_0$ denote the position and velocity estimated errors, respectively. The projections of $e_x(t_0)$-$\dot{e}_x(t_0)$, $e_y(t_0)$-$\dot{e}_y(t_0)$ and $e_z(t_0)$-$\dot{e}_z(t_0)$ of 300 MC runs are represented using the scatters in Figure 13a, Figure 13b and Figure 13c, respectively. In addition, the 3σ ellipses of the 300 estimated errors of the LSM and BiLSTM-LSM methods and the violin plots of each

variable are provided in Figure 13. As seen in Figure 13, the 3σ ellipses of the proposed BiLSTM-LSM are smaller than those of the LSM method, showing that the estimated errors are reduced using the proposed methods.



(**a**) (**b**) (**c**)

Figure 13. Monte Carlo simulation results (when the measurement duration is 6 h). (**a**) Projection of $e_x(t_0)$-$\dot{e}_x(t_0)$; (**b**) projection of $e_y(t_0)$-$\dot{e}_y(t_0)$; (**c**) projection of $e_z(t_0)$-$\dot{e}_z(t_0)$.

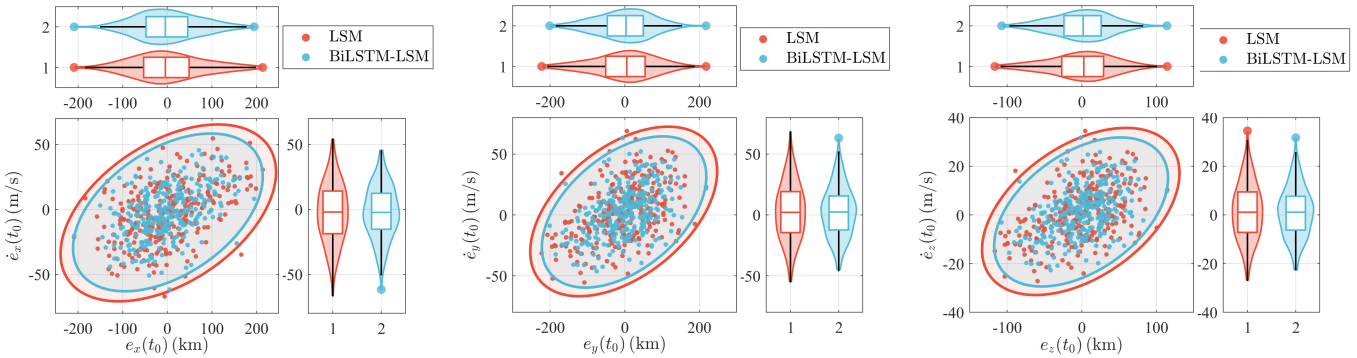
Furthermore, 300 MC simulations are performed for cases with different measurement durations to illustrate the dependency of the method performance on the measurement durations. Five cases, with measurement durations of 2, 3, 4, and 5 h, are simulated. The number of measurements of these five cases are set to 121, 181, 241, and 301, respectively. Two individual metrics, the position-estimated STD σ_R and the velocity-estimated STD σ_V, are defined as

$$\sigma_R = \sqrt{\sigma_x^2(t_0) + \sigma_y^2(t_0) + \sigma_z^2(t_0)}, \tag{26}$$

$$\sigma_V = \sqrt{\sigma_{\dot{x}}^2(t_0) + \sigma_{\dot{y}}^2(t_0) + \sigma_{\dot{z}}^2(t_0)}, \tag{27}$$

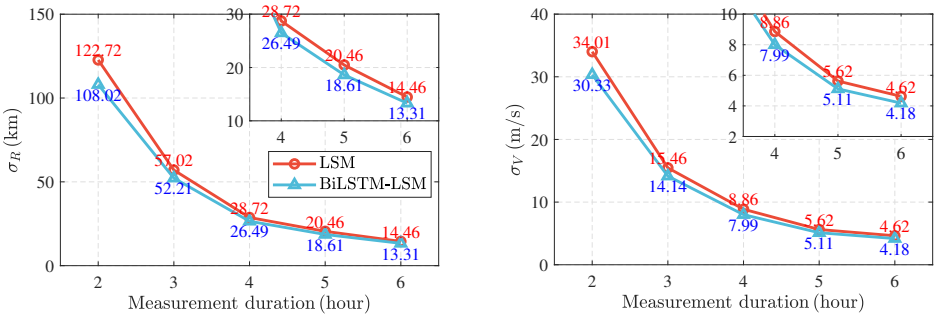
where $\sigma_x(t_0)$, $\sigma_y(t_0)$, $\sigma_z(t_0)$, $\sigma_{\dot{x}}(t_0)$, $\sigma_{\dot{y}}(t_0)$ and $\sigma_{\dot{z}}(t_0)$ are the STDs of $e_x(t_0)$, $e_y(t_0)$, $e_z(t_0)$, $\dot{e}_x(t_0)$, $\dot{e}_y(t_0)$ and $\dot{e}_z(t_0)$ of the 300 MC runs, respectively.

The 300 MC results of the case when the measurement duration is 2 h are shown in Figure 14, and the results for all six cases are compared in Table 12 and Figure 15. The estimation errors decrease as the measurement durations increase. For example, the position-estimated STDs of the two methods are larger than 100 km when the measurement duration is 2 h, whereas the position-estimated STDs are reduced to approximately 14 km when the measurement duration increases to 6 h. Additionally, Figure 15 shows that the OD accuracy is improved using the proposed BiLSTM-LSM method in all six cases.



(**a**) (**b**) (**c**)

Figure 14. Monte Carlo simulation results (when the measurement duration is 2 h). (**a**) Projection of $e_x(t_0)$-$\dot{e}_x(t_0)$; (**b**) projection of $e_y(t_0)$-$\dot{e}_y(t_0)$; (**c**) projection of $e_z(t_0)$-$\dot{e}_z(t_0)$.



(**a**) (**b**)

Figure 15. Monte Carlo simulation results with different measurement durations. (**a**) Position-estimated STD; (**b**) velocity-estimated STD.

Table 12. Estimated STDs comparison results.

Method	Index	Measurement Duration (Hour)				
		2	**3**	**4**	**5**	**6**
BiLSTM-LSM	σ_R (km)	108.0226	52.2082	26.4953	18.6093	13.3092
	σ_V (m/s)	30.3310	14.1457	7.9966	5.1123	4.1821
LSM	σ_R (km)	122.7295	57.0202	28.7276	20.4697	14.4617
	σ_V (m/s)	34.0102	15.4605	8.8635	5.6236	4.6241
LSAR	σ_R (km)	149.8743	69.5296	35.8228	25.4228	18.7784
	σ_V (m/s)	39.0967	18.6796	11.4537	7.0075	6.0896

Moreover, the reduced ratios of the proposed BiLSTM-LSM methods are provided in Figure 16. The reduced ratio is defined as

$$\text{reduced ratio} = \frac{\sigma_i^{\text{LSM}} - \sigma_i^{\text{BiLSTM-LSM}}}{\sigma_i^{\text{LSM}}} \times 100\%, i \in \{R, V\}. \tag{28}$$

As shown in Figure 16, the proposed BiLSTM-LSM method can reduce the estimated errors by approximately 10%. When the measurement duration is 2 h, the reduced ratios in position and velocity are 11.99% and 10.82%, respectively, whereas when the measurement duration is 6 h, the reduced ratios in position and velocity are 7.97% and 9.56%, respectively. Compared with the case of 2 h, the OD improvement performance of the proposed BiLSTM-LSM in the case of 6 h is worse, which is expected since the negative influences of the anomalous measurements are weaker when more measurements are available. However, as shown in Figure 16, the relationship between the measurement duration and the improvement performance is weak. This is because the outputs of the neural network are usually unexplainable, and more results are expected to be analyzed in future work.

Finally, the proposed BiLSTM-LSM is compared with the least sum of absolute residuals (LSAR) method [77], which is capable of rejecting outlier measurements when estimating the states. Different from the LSM (Equations (9) and (10)), the LSAR method obtains the optimal solution by solving a linear programming problem (LPP), given by

$$\begin{array}{ll} \min\limits_{\boldsymbol{b}} & \boldsymbol{c}^T\boldsymbol{b} \\ \text{s.t.} & \boldsymbol{Q}\boldsymbol{b} \leq \boldsymbol{k} \end{array}, \tag{29}$$

where

$$\boldsymbol{Q} = \begin{bmatrix} -\boldsymbol{\Theta} & -\boldsymbol{I}_{mn\times mn} \\ \boldsymbol{\Theta} & -\boldsymbol{I}_{mn\times mn} \end{bmatrix} \in \mathbb{R}^{2mn\times(mn+6)}, \tag{30}$$

$$\boldsymbol{b} = [\delta\hat{\boldsymbol{x}}_0; \boldsymbol{s}] \in \mathbb{R}^{mn+6}, \tag{31}$$

$$\boldsymbol{k} = [-\Delta\boldsymbol{Z}; \Delta\boldsymbol{Z}] \in \mathbb{R}^{2mn}, \tag{32}$$

$$\boldsymbol{c} = \left[\boldsymbol{0}_{6\times 1}; \frac{\boldsymbol{I}_{mn\times 1}}{1.24\sigma}\right] \in \mathbb{R}^{mn+6}. \tag{33}$$

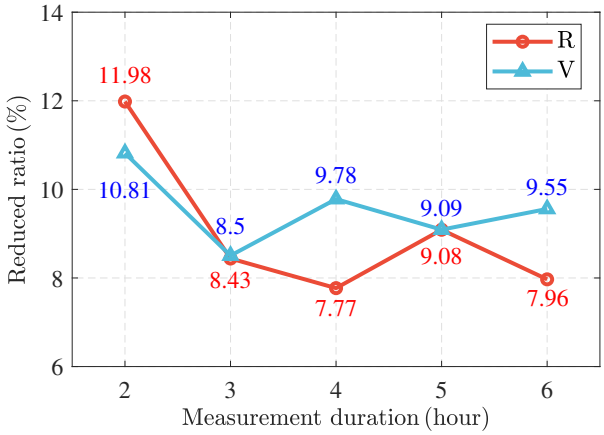


Figure 16. Reduced ratios of using the proposed BiLSTM-LSM.

In this work, the LPP in Equations (29)–(33) are solved using the solver MOSEK (https://www.mosek.com, accessed on 3 September 2024) with its interior point algorithm. The initial state $\delta\hat{\boldsymbol{x}}_0$ is iteratively estimated and the convergence conditions are set as the same as the LSM in Algorithm 1. Three hundred MC runs are carried out for the LSAR method, and the corresponding estimated position and velocity STDs are listed in Table 12. It can be seen from Table 12 that the proposed method obtains more accurate estimations than the LSAR method. The estimations of the LSAR method are even worse than those of the LSM. This is because the LSAR is particularly designed for addressing outlier measurements that have large errors (larger than ten times the measurement STDs). The LSAR performs worse than the standard LSM when none of the measurements are outliers [77] (note that the inaccurate measurements defined in this paper are not outliers), which is, of course, worse than the proposed method.

5. Conclusions

In this paper, a bidirectional long short-term memory (BiLSTM) network-based approach is proposed to improve orbit termination (OD) accuracy. Inaccurate measurements are classified using the designed modified BiLSTM network. After compensating for these inaccurate measurements, a more accurate OD solution is then obtained. The training results show that the overall classification accuracy of the designed modified BiLSTM network is higher than 99%. Compared with the regular unidirectional LSTM networks, the designed modified BiLSTM network has a smaller validation loss and higher predicted accuracy. The proposed method is applied to solve the OD problem of a 4:1 synodic resonant near-rectilinear halo orbit, which is not included in the training and validation dataset. Numerical simulations show that the proposed method can provide a more accurate OD solution than the conventional least square method (LSM), with the estimated errors reduced by 7–12%. Although the LSM is used in this paper, the proposed modified BiLSTM network can be generalized to other systems with different orbit estimation methods by directly replacing the LSM with the corresponding orbit estimation method. In the future, the performances of the proposed method can be validated under different dynamics and measurement models to investigate if such a method can improve OD accuracy in other scenarios. In addition, only Gaussian-distributed measurement errors are considered; thus, it is necessary to further investigate whether the proposed method can improve OD accuracy when non-Gaussian measurement errors (e.g., measurement system errors) exist.

Author Contributions: Methodology, Z.Z. (Zhe Zhang); Writing—review & editing, Y.S.; Supervision, Z.Z. (Zuoxiu Zheng). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 62394353, No. 12150008) and the Beijing Institute of Technology Research Fund Program for lnnovative talents (No. 2022CX01008).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: Thanks to Dong Qiao for providing theoretical guidance.

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

References

1. Abusali, P.A.; Tapley, B.D.; Schutz, B.E. Autonomous navigation of global positioning system satellites using cross-link measurements. *J. Guid. Control Dyn.* **1998**, *21*, 321–327. [CrossRef]
2. Zhou, X.; Qiao, D.; Macdonald, M. Orbit Determination for Impulsively Maneuvering Spacecraft Using Modified State Transition Tensor. *J. Guid. Control Dyn.* **2024**, *47*, 822–839. [CrossRef]
3. Li, X.; Scheeres, D.J.; Qiao, D. Bouncing Return Trajectory Design for Precise Lander Deployment to Asteroids. *J. Guid. Control Dyn.* **2021**, *45*, 121–137. [CrossRef]
4. Zhou, X.; Li, X.; Huo, Z.; Meng, L.; Huang, J. Near-Earth Asteroid Surveillance Constellation in the Sun-Venus Three-Body System. *Space Sci. Technol.* **2022**, *2022*, 9864937. [CrossRef]
5. Duong, N.; Winn, C.B. Orbit determination by range-only data. *J. Spacecr. Rocket.* **1973**, *10*, 132–136. [CrossRef]
6. Hill, K.; Born, G.H. Autonomous interplanetary orbit determination using satellite-to-satellite tracking. *J. Guid. Control Dyn.* **2007**, *30*, 679–686. [CrossRef]
7. Hill, K.; Born, G.H. Autonomous orbit determination from lunar halo orbits using crosslink range. *J. Spacecr. Rocket.* **2008**, *45*, 548–553. [CrossRef]
8. Cinelli, M.; Ortore, E.; Mengali, G.; Quarta, A.A.; Circi, C. Lunar orbits for telecommunication and navigation services. *Astrodynamics* **2024**, *8*, 209–220. [CrossRef]
9. Geller, D.K.; Klein, I. Angles-only navigation state observability during orbital proximity operations. *J. Guid. Control Dyn.* **2014**, *37*, 1976–1983. [CrossRef]
10. Qiao, D.; Zhou, X.; Li, X. Analytical configuration uncertainty propagation of geocentric interferometric detection constellation. *Astrodynamics* **2023**, *7*, 271–284. [CrossRef]
11. Li, X.; Qiao, D.; Circi, C. Mars High Orbit Capture Using Manifolds in the Sun–Mars System. *J. Guid. Control Dyn.* **2020**, *43*, 1383–1392. [CrossRef]
12. Qiao, D.; Zhou, X.; Li, X. Configuration uncertainty propagation of gravitational-wave observatory using a directional state transition tensor. *Chin. J. Aeronaut.* **2024**. [CrossRef]
13. Kedarisetty, S.; Shima, T. Sinusoidal Guidance. *J. Guid. Control Dyn.* **2023**, *47*, 1–16. [CrossRef]
14. Han, H.; Li, X.; Qiao, D. Aerogravity-assist capture into the three-body system: A preliminary design. *Acta Astronaut.* **2022**, *198*, 26–35. [CrossRef]
15. Chen, J.; Qiao, D.; Han, H. Augmented Analytical Aerocapture Guidance by Segmented State Approximation. *J. Guid. Control Dyn.* **2024**, 1–12. [CrossRef]
16. Li, X.; Warier, R.R.; Sanyal, A.K.; Qiao, D. Trajectory Tracking Near Small Bodies Using Only Attitude Control. *J. Guid. Control Dyn.* **2018**, *42*, 109–122. [CrossRef]
17. Li, X.; Qiao, D.; Macdonald, M. Energy-saving capture at mars via backward-stable orbits. *J. Guid. Control Dyn.* **2019**, *42*, 1136–1145. [CrossRef]
18. Yao, Q.; Han, H.; Qiao, D. Nonsingular Fixed-Time Tracking Guidance for Mars Aerocapture With Neural Compensation. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3686–3696. [CrossRef]
19. Luo, Y.; Qin, T.; Zhou, X. Observability Analysis and Improvement Approach for Cooperative Optical Orbit Determination. *Aerospace* **2022**, *9*, 166. [CrossRef]
20. Qiao, D.; Zhou, X.; Zhao, Z.; Qin, T. Asteroid Approaching Orbit Optimization Considering Optical Navigation Observability. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 5165–5179. [CrossRef]
21. Fossà, A.; Losacco, M.; Armellin, R. Perturbed initial orbit determination. *Astrodynamics* **2024**, 1–16. [CrossRef]
22. Cao, Z.; Li, X.; Qi, Y. Semi-Analytical Assessment of Multiple Gravity-Assist Opportunities Based on Feasibility Domains. *J. Guid. Control Dyn.* **2024**, *47*, 1645–1659. [CrossRef]
23. Zhou, X.; Qin, T.; Macdonald, M.; Qiao, D. Observability analysis of cooperative orbit determination using inertial inter-spacecraft angle measurements. *Acta Astronaut.* **2023**, *210*, 289–302. [CrossRef]
24. Park, R.S.; Scheeres, D.J. Nonlinear semi-analytic methods for trajectory estimation. *J. Guid. Control Dyn.* **2007**, *30*, 1668–1676. [CrossRef]

25. Zhang, S.; Fu, T.; Chen, D.; Ding, S.; Gao, M. An Initial Orbit Determination Method Using Single-Site Very Short Arc Radar Observations. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 1856–1872. [CrossRef]
26. Zhou, X.; Jia, F.; Li, X. Configuration Stability Analysis for Geocentric Space Gravitational-Wave Observatories. *Aerospace* **2022**, *9*, 519. [CrossRef]
27. Kaufman, E.; Lovell, T.A.; Lee, T. Nonlinear observability for relative orbit determination with angles-only measurements. *J. Astronaut. Sci.* **2016**, *63*, 60–80. [CrossRef]
28. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422. [CrossRef]
29. Chang, L.; Hu, B.; Li, A.; Qin, F. Transformed unscented kalman filter. *IEEE Trans. Autom. Control* **2013**, *58*, 252–257. [CrossRef]
30. Arasaratnam, I.; Haykin, S. Cubature kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269. [CrossRef]
31. Jia, B.; Xin, M.; Cheng, Y. High-degree cubature Kalman filter. *Automatica* **2013**, *49*, 510–518. [CrossRef]
32. Zhang, L.; Yang, H.; Lu, H.; Zhang, S.; Cai, H.; Qian, S. Cubature Kalman filtering for relative spacecraft attitude and position estimation. *Acta Astronaut.* **2014**, *105*, 254–264. [CrossRef]
33. Adurthi, N.; Singla, P. Conjugate unscented transformation-based approach for accurate conjunction analysis. *J. Guid. Control Dyn.* **2015**, *38*, 1642–1658. [CrossRef]
34. Nanda, A.; Singla, P.; Karami, M.A. Conjugate unscented transformation–based uncertainty analysis of energy harvesters. *J. Intell. Mater. Syst. Struct.* **2018**, *29*, 3614–3633. [CrossRef] [PubMed]
35. Luo, Y.; Yang, Z. A review of uncertainty propagation in orbital mechanics. *Prog. Aerosp. Sci.* **2017**, *89*, 23–39. [CrossRef]
36. Qiao, D.; Zhou, X.; Li, X. Feasible domain analysis of heliocentric gravitational-wave detection configuration using semi-analytical uncertainty propagation. *Adv. Space Res.* **2023**, *72*, 4115–4131. [CrossRef]
37. Shang, H.; Liu, Y. Assessing accessibility of main-belt asteroids based on Gaussian process regression. *J. Guid. Control Dyn.* **2017**, *40*, 1144–1154. [CrossRef]
38. Sun, Z.; Simo, J.; Gong, S. Satellite Attitude Identification and Prediction Based on Neural Network Compensation. *Space Sci. Technol.* **2023**, *3*, 9. [CrossRef]
39. Xu, L.; Zhang, G.; Qiu, S.; Cao, X. Optimal Multi-impulse Linear Rendezvous via Reinforcement Learning. *Space Sci. Technol.* **2023**, *3*, 47. [CrossRef]
40. Li, H.; Chen, S.; Izzo, D.; Baoyin, H. Deep networks as approximators of optimal low-thrust and multi-impulse cost in multitarget missions. *Acta Astronaut.* **2020**, *166*, 469–481. [CrossRef]
41. Gao, T.; Peng, H.; Bai, X. CaLibration of atmospheric density model based on Gaussian Processes. *Acta Astronaut.* **2020**, *168*, 273–281. [CrossRef]
42. Yang, B.; Li, S.; Feng, J.; Vasile, M. Fast Solver for J2-Perturbed Lambert Problem Using Deep Neural Network. *J. Guid. Control Dyn.* **2022**, *45*, 875–884. [CrossRef]
43. Peng, H.; Bai, X. Improving orbit prediction accuracy through supervised machine learning. *Adv. Space Res.* **2018**, *61*, 2628–2646. [CrossRef]
44. Peng, H.; Bai, X. Artificial neural network-based machine learning approach to improve orbit prediction accuracy. *J. Spacecr. Rocket.* **2018**, *55*, 1248–1260. [CrossRef]
45. Peng, H.; Bai, X. Gaussian Processes for improving orbit prediction accuracy. *Acta Astronaut.* **2019**, *161*, 44–56. [CrossRef]
46. Peng, H.; Bai, X. Comparative evaluation of three machine learning algorithms on improving orbit prediction accuracy. *Astrodynamics* **2019**, *3*, 325–343. [CrossRef]
47. Peng, H.; Bai, X. Fusion of a machine learning approach and classical orbit predictions. *Acta Astronaut.* **2021**, *184*, 222–240. [CrossRef]
48. Peng, H.; Bai, X. Machine Learning Approach to Improve Satellite Orbit Prediction Accuracy Using Publicly Available Data. *J. Astronaut. Sci.* **2020**, *67*, 762–793. [CrossRef]
49. Li, B.; Huang, J.; Feng, Y.; Wang, F.; Sang, J. A Machine Learning-Based Approach for Improved Orbit Predictions of LEO Space Debris With Sparse Tracking Data From a Single Station. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 4253–4268. [CrossRef]
50. Zhou, X.; Qin, T.; Meng, L. Maneuvering Spacecraft Orbit Determination Using Polynomial Representation. *Aerospace* **2022**, *9*, 257. [CrossRef]
51. Zhou, X.; Qin, T.; Ji, M.; Qiao, D. A LSTM assisted orbit determination algorithm for spacecraft executing continuous maneuver. *Acta Astronaut.* **2022**, *204*, 568–582. [CrossRef]
52. Zhou, X.; Qiao, D.; Li, X. Neural Network-Based Method for Orbit Uncertainty Propagation and Estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 1176–1193. [CrossRef]
53. Zhou, X.; Wang, S.; Qin, T. Multi-Spacecraft Tracking and Data Association Based on Uncertainty Propagation. *Appl. Sci.* **2022**, *12*, 7660–7671. [CrossRef]
54. Zhou, X.; Cheng, Y.; Qiao, D.; Huo, Z. An adaptive surrogate model-based fast planning for swarm safe migration along halo orbit. *Acta Astronaut.* **2022**, *194*, 309–322. [CrossRef]
55. Zhou, X.; Li, X.; Zhang, Z. Neural Network–Assisted Initial Orbit Determination Method for Libration Point Orbits. *J. Aerosp. Eng.* **2024**, *37*, 4024046. [CrossRef]
56. Schuster, M.; Paliwal, K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
57. Bonet, I.; Rodriguez, A.; Grau, I. Bidirectional Recurrent Neural Networks for Biological Sequences Prediction BT—Advances in Soft Computing and Its Applications; Springer: Berlin/Heidelberg, Germany, 2013; pp. 139–149.

58. Gouhara, K.; Watanabe, T.; Uchikawa, Y. Learning process of recurrent neural networks. In Proceedings of the 1991 IEEE International Joint Conference on Neural Networks, Singapore, 18–21 November 1991; Volume 1, pp. 746–751. [CrossRef]
59. Boden, M.; Hawkins, J. Improved access to sequential motifs: A note on the architectural bias of recurrent networks. *IEEE Trans. Neural Netw.* **2005**, *16*, 491–494. [CrossRef] [PubMed]
60. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
61. Greff, K.; Srivastava, R.K.; Koutnik, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [CrossRef]
62. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. The Performance of LSTM and BiLSTM in Forecasting Time Series. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 3285–3292. [CrossRef]
63. Hameed, Z.; Garcia-Zapirain, B. Sentiment Classification Using a Single-Layered BiLSTM Model. *IEEE Access* **2020**, *8*, 73992–74001. [CrossRef]
64. Bretagnon, P. Theory for the motion of all the planets-The VSOP82 solution. *Astron. Astrophys.* **1982**, *114*, 278–288.
65. Moisson, X.; Bretagnon, P. Analytical planetary solution VSOP2000. *Celest. Mech. Dyn. Astron.* **2001**, *80*, 205–213. [CrossRef]
66. Han, H.; Li, X.; Ren, J. Transfer between Libration orbits through the outer branches of manifolds for Phobos exploration. *Acta Astronaut.* **2021**, *189*, 321–336. 10.1016/j.actaastro.2021.08.035 [CrossRef]
67. Ou, Y.; Zhang, H.; Li, B. Absolute orbit determination using line-of-sight vector measurements between formation flying spacecraft. *Astrophys. Space Sci.* **2018**, *363*, 1–13. [CrossRef]
68. Hu, Y.; Sharf, I.; Chen, L. Three-spacecraft autonomous orbit determination and observability analysis with inertial angles-only measurements. *Acta Astronaut.* **2020**, *170*, 106–121. [CrossRef]
69. Hu, Y.; Sharf, I.; Chen, L. Distributed orbit determination and observability analysis for satellite constellations with angles-only measurements. *Automatica* **2021**, *129*, 109626. [CrossRef]
70. Karim, F.; Majumdar, S.; Darabi, H.; Chen, S. LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access* **2018**, *6*, 1662–1669. [CrossRef]
71. Daniel, J.G. Generating Periodic Orbits in the Circular Restricted Three-Body Problem with Applications to Lunar South Pole Coverage. Ph.D. Thesis, Purdue University, West Lafayette, IN, USA, 2006.
72. Zhou, X.; Qiao, D.; Li, X. Adaptive Order-Switching Kalman Filter for Orbit Determination Using Deep-Neural-Network-Based Nonlinearity Detection. *J. Spacecr. Rocket.* **2023**, *60*, 1724–1741. [CrossRef]
73. Howell, K.C.; Pernicka, H.J. Numerical determination of Lissajous trajectories in the restricted three-body problem. *Celest. Mech.* **1987**, *41*, 107–124. [CrossRef]
74. Dormand, J.R.; Prince, P.J. A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.* **1980**, *6*, 19–26. [CrossRef]
75. Olsson, A.; Sandberg, G.; Dahlblom, O. On Latin hypercube sampling for structural reliability analysis. *Struct. Saf.* **2003**, *25*, 47–68. [CrossRef]
76. Ghiasi-Shirazi, K. Competitive Cross-Entropy Loss: A Study on Training Single-Layer Neural Networks for Solving Nonlinearly Separable Classification Problems. *Neural Process. Lett.* **2019**, *50*, 1115–1122. [CrossRef]
77. Prabhu, K.; Majji, M.; Alfriend, K.T. Least Sum of Absolute Residuals Orbit Determination. *J. Guid. Control Dyn.* **2022**, *45*, 468–480. [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.