

Article

A Complement Method for Magnetic Data Based on TCN-SE Model

Wenqing Chen ^{1,2}, Rui Zhang ^{2,*}, Chenguang Shi ^{1,2} , Ye Zhu ¹ and Xiaodong Lin ¹

¹ Innovation Academy for Microsatellites of Chinese Academy of Sciences, Shanghai 201203, China; cc123456@mail.ustc.edu.cn (W.C.); icescg@mail.ustc.edu.cn (C.S.); zhuye@microsate.com (Y.Z.); linxd@microsate.com (X.L.)

² University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: zhangr@microsate.com

Abstract: The magnetometer is a vital measurement component for attitude measurement of near-Earth satellites and autonomous magnetic navigation, and monitoring health is significant. However, due to the compact structure of the microsatellites, the stray magnetic changes caused by the complex working conditions of each system will inevitably interfere with the magnetometer measurement. In addition, due to the limited capacity of the satellite–ground measurement channels and the telemetry errors caused by the harsh space environment, the magnetic data collected by the ground station are partially missing. Therefore, reconstructing the telemetry data on the ground has become one of the key technologies for establishing a high-precision magnetometer twin model. In this paper, firstly, the stray magnetic interference is eliminated by correcting the installation matrix for different working conditions. Then, the autocorrelation characteristics of the residuals are analyzed, and the TCN-SE (temporal convolutional network-squeeze and excitation) network with long-term memory is designed to model and extrapolate the historical residual data. In addition, *MAE* (mean absolute error) is used to analyze the data without missing at the corresponding time in the forecast period and decreases to 74.63 nT. The above steps realize the accurate mapping from the simulation values to the actual values, thereby achieving the reconstruction of missing data and establishing a solid foundation for the judgment of the health state of the magnetometer.

Keywords: magnetometer; actual magnetic data; stray magnetic; different working conditions; TCN-SE network



Citation: Chen, W.; Zhang, R.; Shi, C.; Zhu, Y.; Lin, X. A Complement Method for Magnetic Data Based on TCN-SE Model. *Sensors* **2022**, *22*, 8277. <https://doi.org/10.3390/s22218277>

Academic Editors: Gianni D'Angelo and Arcangelo Castiglione

Received: 28 August 2022

Accepted: 13 October 2022

Published: 28 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

Geomagnetic navigation is a critical autonomous navigation method for near-Earth spacecraft, which has strong real-time, concealment, and interference suppression. Geomagnetic navigation uses the three-axis magnetometer installed on the spacecraft to measure the geomagnetic intensity in three directions that are not coplanar in space below 1000 km, and compares it with the nominal geomagnetic model to determine the orbit and attitude of the satellite [1]. Therefore, the health status of magnetometers is very important for satellite navigation and attitude determination. However, due to the limited capacity of the satellite–Earth measurement channels and the telemetry errors caused by the harsh space environment, the magnetic data are often missing, which makes it difficult to study the laws of magnetic data and establish accurate models using historical data. Thus, establishing an accurate complement model is a prerequisite for building a geomagnetic model.

The international geomagnetic reference model (IGRF) [2] and the world geomagnetic field model (WMM) are the commonly used geomagnetic models. They are based on the Gaussian spherical harmonic analysis theory, and building the expression between the geomagnetic vector elements and the geographical location. Then, the researchers comprehensively utilize the observation data from the marine magnetic survey, ground station magnetic survey, aeromagnetic survey, and satellite magnetic survey to solve

the Gaussian coefficients of the model based on the standard least squares method. Although these models can comprehensively describe the distribution characteristics of the geomagnetic field, they ignore the local details such as the external field of the geomagnetic field and lithosphere. In addition, the geomagnetic measurement is affected by the stray current in different directions emitted by satellite components under various working conditions. Accordingly, the missing values cannot be directly supplemented by the IGRF and WMM models.

Marco et al. [3] predicted the subsequent missing points from the data before the missing points by the autoregressive moving average model, but this does not make full use of the subsequent data of the missing points and the overall utilization of the data is low. Zhu et al. [4] used the SVM algorithm to predict missing data, however, the SVM method cannot fill in a large number of missing data effectively. Considering the strong nonlinear fitting ability of the neural network, the authors of [5] established the BP neural network to predict the lost data and had effective results for filling the missing value. However, the BP network is not a time series model and it did not perform feature extraction and signal-to-noise ratio improvement, which is difficult to achieve high precision requirements. Li et al. [6] obtained the probability distribution of missing data by analyzing the statistical characteristics, and used the maximum likelihood to estimate the most suitable values.

Due to the periodic rotation of the satellite around the Earth, the data sampled by the sensors often have autocorrelation. Therefore, it is a better way to adopt the time series models with historical memory as the prediction model.

At present, the traditional autoregressive prediction technology has been mature, including autoregressive moving average (ARMA) [Baptista M] and autoregressive integrated moving average (ARIMA) [7,8]. These statistical models are widely used in the industry. However, they can only be linearly expressed and the signal must be assumed to be stationary. The Kalman filter [9,10], least squares support vector machine (LS-SVM) [11,12], and other methods have been studied for a long time and the theories are mature. Therefore, they have been applied in many fields to solve time series prediction tasks. However, these methods can only deal with short-term autocorrelation tasks and are difficult to model long-term dependence of time series.

In recent years, with the rapid development of deep learning, the time series prediction method based on deep learning theory has been highly studied.

The authors of [13] used historical information of time series to predict future Turkish electricity load by a recurrent neural network (RNN). The authors of [14] adopted long short-term memory (LSTM) as a one-step prediction model for satellite telemetry data, and proposed an adaptive threshold algorithm to obtain the best threshold. It achieves telemetry data anomaly detection finally. The authors of [15] proposed a model combining a two-channel convolutional neural network (CNN) and LSTM. They used this model to predict short-term photovoltaic power and achieved better results. However, LSTM has the disadvantages of difficult model training, long training time, and unstable gradients in long-term dependence modeling. Temporal convolutional network [16] is a one-dimensional convolutional neural network. Compared with LSTM, TCN has more stable gradients, a flexible receptive field, and a shorter training time. Studies have shown that the network has achieved better results in audio synthesis [17], word language modeling [18], machine translation [19], and other fields.

In this paper, considering stray magnetic fields in different directions under various working conditions, the linear model is first used to correct the installation matrix. Moreover, since the residuals between the actual value and corrected simulation values in the installation coordinate system have autocorrelation, and to avoid the establishment of complex physical models and a large number of calculations, establishing a temporal model with historical memory ability is an effective and practical method that conforms to the residual characteristics.

2. Data Description and Calibrating Installation Matrix

2.1. Coordinate Systems

Before showing the data and calibrating method, we introduce several coordinate systems involved in this paper.

1. J2000 coordinate system: As shown in Figure 1, the origin is at the center of the Earth, the X axis points to the vernal equinox, the Y axis is in the equatorial plane and perpendicular to the X axis, and the Z axis points to the North Pole.
2. Orbital coordinate system (VVLH): The origin of the orbital coordinate system is located at the satellite centroid, the Z axis points to the Earth center, the X axis is in the orbital plane and perpendicular to the Z axis, and the Y axis is determined by the right-hand rule.
3. Body coordinate system: The origin of the body coordinate is located at the satellite's center of mass, the X axis points to the satellite's flight direction, the Z axis is in the satellite's longitudinal symmetry plane, and the Y axis is perpendicular to the satellite's longitudinal symmetry plane.
4. Installation coordinate system: The installation position of the magnetometer varies from satellite to satellite. The three-axis installation coordinate system of the magnetometer is obtained by multiplying the body coordinate system by the installation matrix. The coordinate system is shown in Figure 2.

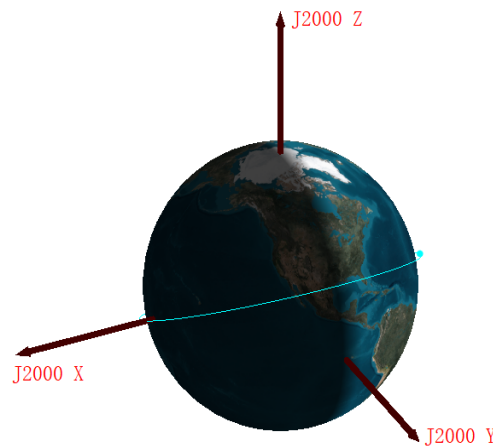


Figure 1. J2000 coordinate system. The cyan line is the satellite orbit.

2.2. Geomagnetic Field Measurement

The geomagnetic field measurement can be expressed as,

$$\begin{aligned} M_A &= (R_{AB} * R_{BO} * R_{OI}) * M_{IGRF} + \tilde{\xi} \\ &= (\hat{R}_{AB} + \tilde{R}_{AB}) * (\hat{R}_{BO} + \tilde{R}_{BO}) * (\hat{R}_{OI} + \tilde{R}_{OI}) * (M_{IGRF} + \tilde{M}_{IGRF}) + \tilde{\xi} \end{aligned} \quad (1)$$

where M_A and M_{IGRF} are the magnetic vectors under the installation coordinate system (the fluxgate magnetometer measurement coordinate system) and J2000 inertial coordinate system (obtained by the IGRF model, which is established in Satellite Tool Kit), respectively. $\tilde{M}_{IGRF}$ is the influence of the lithosphere and external field, which is not considered in the IGRF model. R_{OI} , R_{BO} , and R_{AB} are the transformation matrices of the inertial coordinate system to the orbital coordinate system, orbital coordinate system to body coordinate system, and body coordinate system to the installation coordinate system, respectively. $\hat{R}_{AB}$, $\hat{R}_{BO}$, and $\hat{R}_{OI}$ are the main parts of the three which are evaluated at the ground station and $\tilde{R}_{AB}$, $\tilde{R}_{BO}$, and $\tilde{R}_{OI}$ are the errors. $\tilde{\xi}$ is the error term. Since the orbit error is less than

10 m and the attitude determination precision of the star tracker is on the order of angular seconds [20], the R_{BO} and R_{OI} can be ignored. The above formula can be simplified as,

$$M_A = (\hat{R}_{AB} * \hat{R}_{BO} * \hat{R}_{OI} + \tilde{R}_{AB} * \hat{R}_{BO} * \hat{R}_{OI}) * (M_{IGRF} + \tilde{M}_{IGRF}) + \tilde{\xi} \quad (2)$$

where $\hat{R}_{AB}$, $\hat{R}_{BO}$, and $\hat{R}_{OI}$ are not affected by working conditions and are the determined matrices. Excluding the considered part, the residual error M_{RS} (as shown in Figure 3) is,

$$\begin{aligned} M_{RS} &= M_A - \hat{R}_{AB} * \hat{R}_{BO} * \hat{R}_{OI} * M_{IGRF} \\ &= \hat{R}_{AB} * \hat{R}_{BO} * \hat{R}_{OI} * \tilde{M}_{IGRF} + \tilde{R}_{AB} * \hat{R}_{BO} * \hat{R}_{OI} * M_{IGRF} \\ &\quad + \tilde{R}_{AB} * \hat{R}_{BO} * \hat{R}_{OI} * \tilde{M}_{IGRF} + \tilde{\xi} \end{aligned} \quad (3)$$

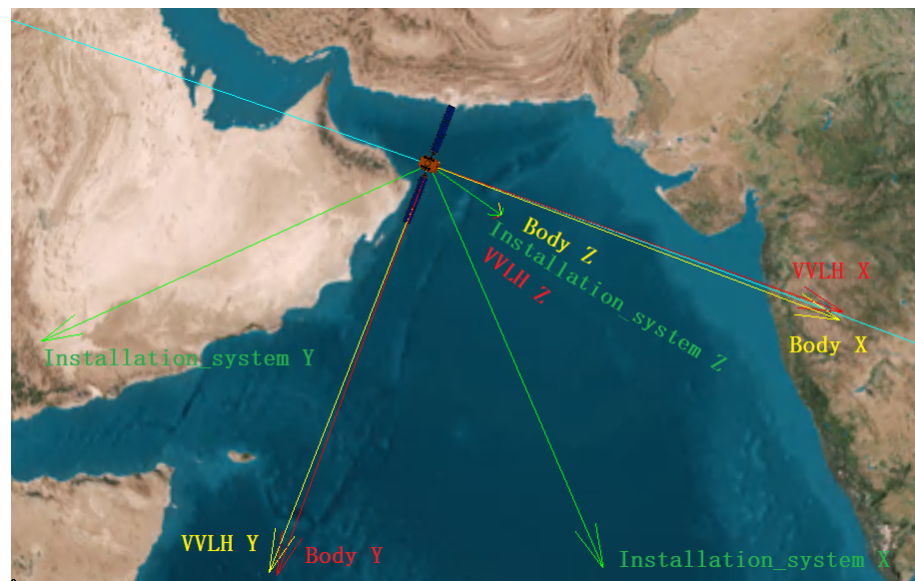


Figure 2. Orbital coordinate system, body coordinate system, and magnetometer installation coordinate system.

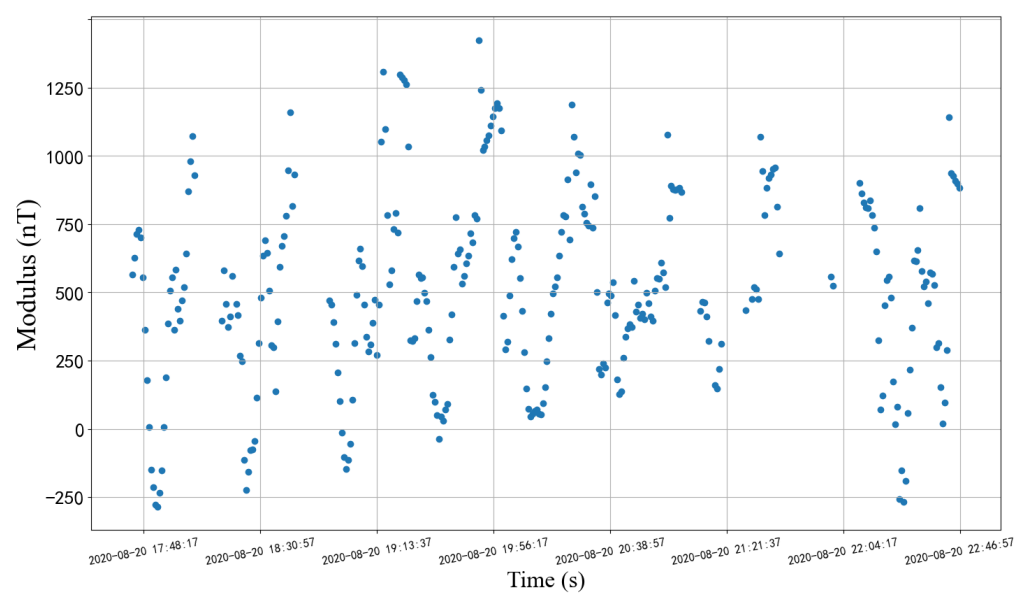


Figure 3. The residuals between the actual data and the simulation data in the installation coordinate system correspond to the subfigure in Figure 4.

The sources of residual error are as follows. The multiplicative or additive combination of these influences constitutes the residual error.

1. Satellite body components: The current of solar panel, battery and magnetic torque, and hard and soft magnetic materials on the satellite generate magnetic fields [21]. These disturbances change with attitude, showing a time-related regularity. This part is the source of stray magnetism.
2. External magnetic interference (not included in IGRF model): (1) The local anomaly of the lithosphere: it mainly comes from the short wavelength magnetic field caused by the geological characteristics of the Earth's surface (mountains, oceans, minerals, etc.) [22]. (2) External field: the space current system above the Earth's surface, which is mainly distributed in the ionosphere, that is, the upper atmosphere ionized by solar high-energy radiation and cosmic ray excitation [23]. The impact of this item can reach dozens of nT [24]. These two terms show periodic changes with orbital periods.
3. IGRF fitting error. This part is minimal. Ref. [25] shows that the accuracy of the spherical harmonic coefficient of the eighth-order IGRF-11 is 0.1 nT.
4. Effects of temperature change: (1) Uneven expansion of the satellite and the changes in the installation coordinate system, which has a slight impact. (2) Thermal drift, which is relatively large. Ref. [26] shows that the maximum drift of the magnetometer reaches 200 nT between 0 and 40 °C. This effect of temperature changes regularly with the satellite in the Earth's shadow area or sunshine area.

2.3. Data Description

The data processed in this paper are all three-axis vector moduli of the geomagnetic field. In this paper, a total of 10 days of on-orbit magnetic data are collected and 5722 discrete points are obtained by a sampling frequency of 128 s. Figure 4 shows the actual sampled data, and the enlarged subfigure shows part of the modulus of the actual data and the simulation data in the installation coordinate system (before calibrating). The overall varying ranges are [15,492.92 nT, 42,431.39 nT] and [17,769.52 nT, 44,356.67 nT], respectively.

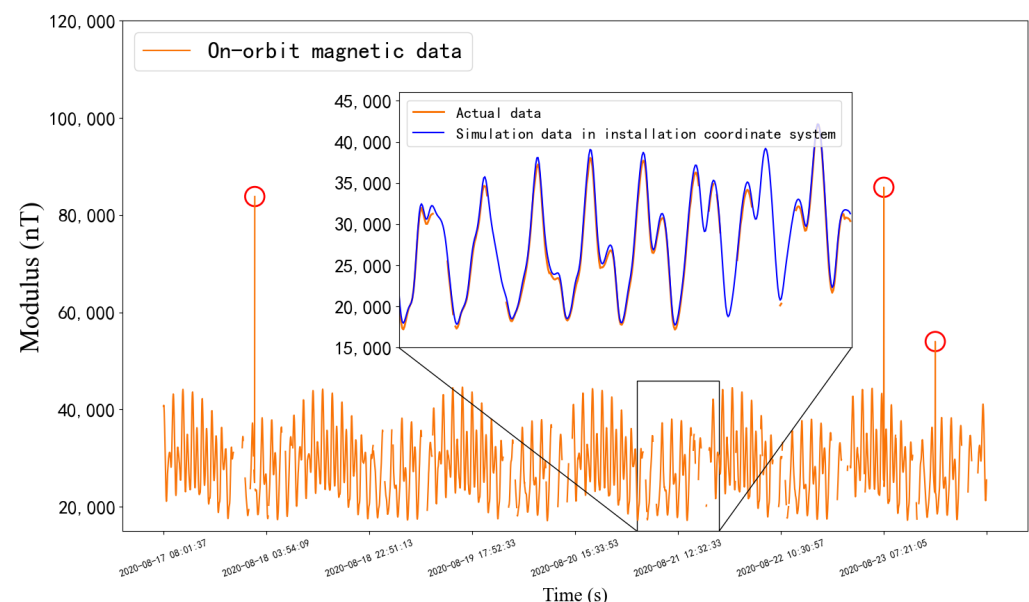


Figure 4. The father graph shows the on-orbit magnetic data. We select one segment for showing details of the actual data and the simulation data in the installation coordinate system, which are labeled in orange and blue curves, respectively.

Because of the complex space environment, remote sensing errors may happen, which results in abnormal values marked in red circles, as shown in Figure 4. The steps for handling such outliers are as follows,

1. Use the method in Algorithm 1 to detect abnormal sampling points:
2. The outliers detected in step 1 are deleted, and then supplemented by our imputation method proposed in Figure 5.

Algorithm 1: 3σ outlier detection method

```

for sample point  $M_i$  do
   $\mu$  and  $\sigma$  are the mean value and the variance of all the sampling points.
  if  $M_i > \mu + 3 * \sigma$  or  $M_i < \mu - 3 * \sigma$  then
    |  $M_i$  is outlier;
  end
end
  
```

Moreover, due to the missing points caused by communication blocking, 505 points are missed, and the maximum missing interval is 24 points, which will provide a basis for the following model parameter setting.

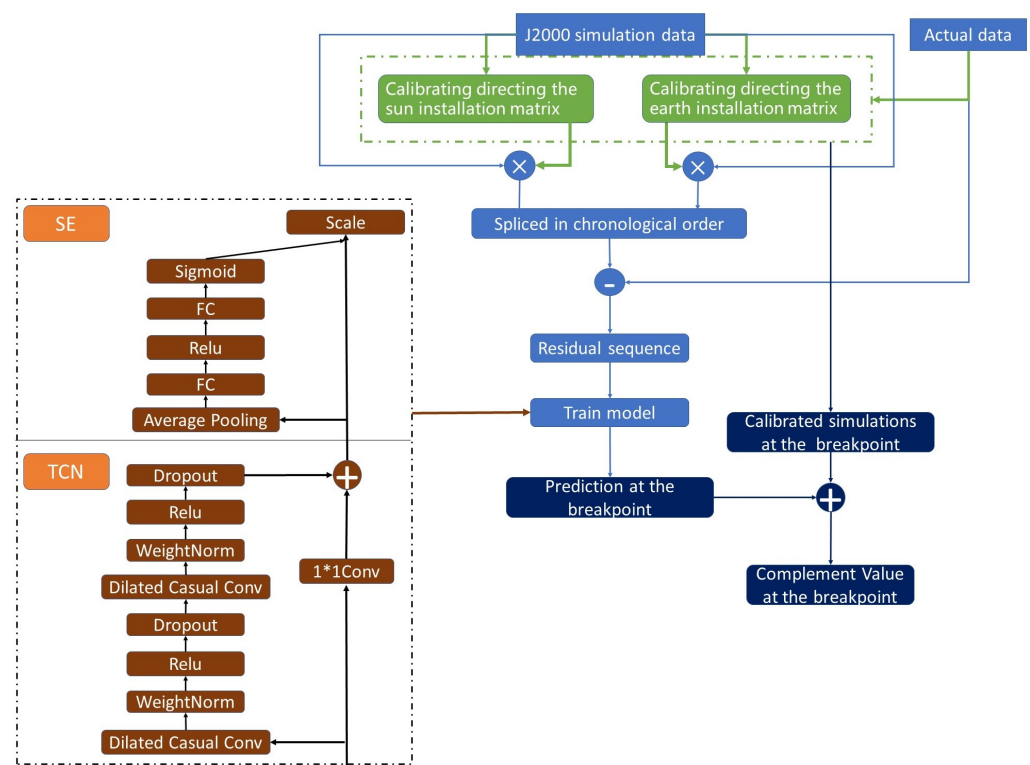


Figure 5. Complement algorithm flow chart.

Figure 3 shows partial residual errors described in Formula (3). The overall range is $[-537.97 \text{ nT}, 1644.46 \text{ nT}]$. Ref. [27] used the extended Kalman filtering method to simulate the real magnetic measurement data of ERBS and GRO satellites, and the orbit determination error was 10–40 km under the condition of measurement error of 400 nT. Based on engineering experience, the error range obviously cannot meet the engineering requirements.

When the satellite directs the Earth and the Sun, respectively, the satellite body temperature is different, resulting in different degrees of thermal deformation and different effects on R_{AB} and $\tilde{\xi}$. At the same time, the difference in satellite attitude makes the stray magnetism in different directions have different effects on $\tilde{\xi}$ under the installation system. Therefore, we first deal with the additive and multiplicative errors caused by working conditions in Formula (3).

2.4. Calibration of Installation Matrix

The corresponding times of directing the Sun and the Earth are obtained from the satellite orbit model established in STK. As for the time in the conversion period, it only accounts for 0.17% in one period, and the corrected simulation values in the installation coordinate system of the conversion period by calibrating directing Sun installation matrix do not generate large errors. So we will not calibrate separately. In this paper, three multi-variate linear regression models are used to correct the installation matrix corresponding to the X-Y-Z three-axis vectors, and the overall format is (matrix format),

$$\begin{bmatrix} \hat{M}_{A_X} \\ \hat{M}_{A_Y} \\ \hat{M}_{A_Z} \end{bmatrix} = \hat{R}_{AB}' * \hat{R}_{AB} * \hat{R}_{BO} * \hat{R}_{OI} * \begin{bmatrix} M_{IGRF_X} \\ M_{IGRF_Y} \\ M_{IGRF_Z} \end{bmatrix} + \begin{bmatrix} \xi_X \\ \xi_Y \\ \xi_Z \end{bmatrix} \quad (4)$$

where $[\hat{M}_{A_X}, \hat{M}_{A_Y}, \hat{M}_{A_Z}]^T$ are the actual value of the three-axis magnetic vector, $[M_{IGRF_X}, M_{IGRF_Y}, M_{IGRF_Z}]$ are the three-axis magnetic vector under the inertial system, and $[\xi_X, \xi_Y, \xi_Z]$ are the error vectors. All the above vectors are one-dimensional data of $1 \times n$, and n is the number of actual values. The $\hat{R}_{AB}'$ is the calibrated installation matrix (3×3) from the body system to the installation system, and $[\hat{R}_{AB}, \hat{R}_{BO}, \hat{R}_{OI}]$ are equal to the ones in use and are all sizes of (3×3). The three linear regression models follow the assumptions:

1. The three-axis magnetic vectors $[M_{IGRF_X}, M_{IGRF_Y}, M_{IGRF_Z}]$ are not related to each other;
2. The $[\xi_X, \xi_Y, \xi_Z]$ are independent and identically distributed;
3. The three-axis vectors $[M_{IGRF_X}, M_{IGRF_Y}, M_{IGRF_Z}]$ are not related to the random term $[\xi_X, \xi_Y, \xi_Z]$, that is, $Cov(M_{IGRF_ij}, \xi_i) = 0, i \in [X, Y, Z], j = 1, 2, \dots, n$.

Since there is no analytical solution when $[M_{IGRF_X}, M_{IGRF_Y}, M_{IGRF_Z}] \times [M_{IGRF_X}, M_{IGRF_Y}, M_{IGRF_Z}]^T$ is not an invertible matrix, it takes a lot of time when the number of data increases, especially when calculating the inverse matrix. So we use SGD (stochastic gradient descent). The SGD algorithm is an improved algorithm based on gradient descent. It randomly selects a sample to iterate once, rather than for all samples. Therefore, Algorithm 2 significantly reduces the calculation magnitude.

Algorithm 2: The SGD algorithm (take the X axis as an example)

```

Three regression coefficients corresponding to  $\hat{R}_{AB}'$  in  $\hat{M}_{A\_X}$ :
for  $i = [1, 2, 3]$  do
    while  $epochs < total\ iterations\ n$  do
        Randomly select a number  $j$  from 1 to  $n$ ;
         $\beta_{ij} = \beta_{ij} - \alpha * (\hat{M}_{A\_X} - M_{A\_X}) * M_{IGRF\_X}$ 
    end
end
 $\xi_X$ : while  $epochs < total\ iterations\ n$  do
    Randomly select a number  $j$  from 1 to  $n$ ;
     $\xi_X = \xi_X - \alpha * (\hat{M}_{A\_X} - M_{A\_X}) * M_{IGRF\_X}$ 
end

```

The data sampled is divided into the training set and the test set by the ratio of 8:2. The training data are used to fit the installation matrix. The test set is used to test the calibration effect.

After correcting the installation matrix by the above steps, the residual range between the actual value and the simulation value translated to the installation coordinate system is in $[-540.55\text{ nT}, 511.98\text{ nT}]$ in the test set. It can be seen from Figure 6 that this step

significantly eliminates the influence of different working conditions and greatly reduces the error range.

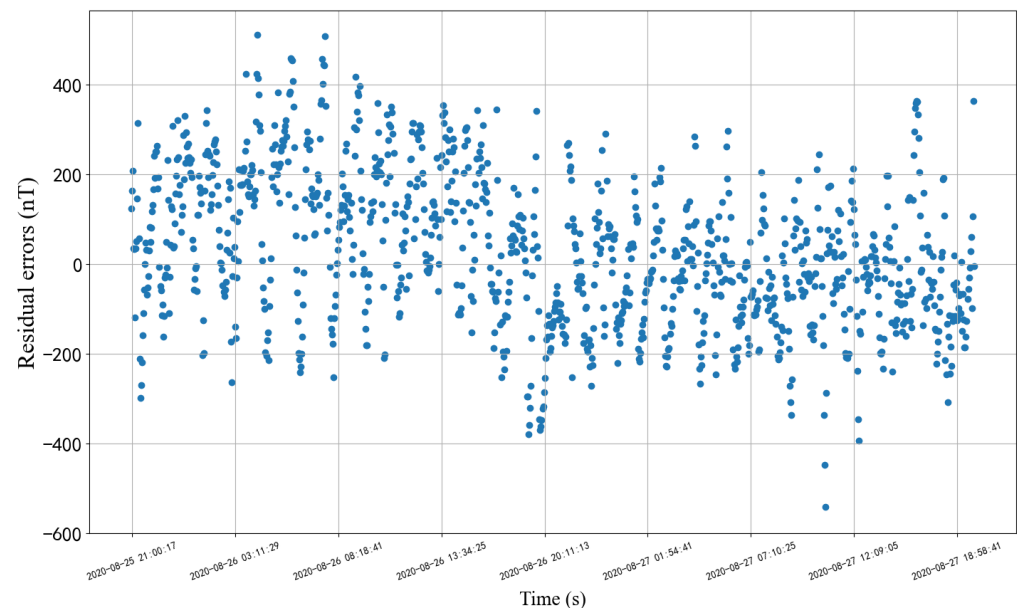


Figure 6. Residuals after calibrating installation matrix.

However, the linear regression only assumes the linear relationship between the dependent variable and the independent variable, and cannot directly deal with the multiplicative errors described in Section 2.2 (these errors also cannot be expressed by mathematical formulas with clear physical meaning) and high-order small quantities. In addition, the influence of local lithospheric anomaly and external field are not included in the IGRF model, and this step cannot deal with this impact. Therefore, we consider establishing a nonlinear model to further reduce the error by learning the rules of these residuals by using the characteristics of the residual sequence.

3. Processing of Residual Errors

3.1. Autocorrelation of Residual Sequence

The residual sequences of different working conditions after calibration are obtained from Section 2.4, and then spliced in chronological order. Then the autocorrelation of the observation sequence is calculated by using the autocorrelation function. The autocorrelation function is defined as,

$$\rho_k = \frac{\sum_{t=1}^{n-k} (RS_t - \bar{RS})(RS_{t-k} - \bar{RS})}{\sum_{t=1}^n (RS_t - \bar{RS})^2} \quad (5)$$

where RS_t is the t -th value of residual sequence RS , $\bar{RS}$ is the mean value of sequence RS , k is the lag order, and n is the sequence length.

Figure 7 shows the calculation result. The dashed line represents the 99% confidence intervals. When the autocorrelation value corresponds to a lag order located outside the dashed line, it indicates that the lag order has autocorrelation in statistical significance. It can be seen from the figure that the residual sequence remains the maximum 135-order autocorrelation (since the subsequent orders do not exceed the confidence intervals, only the first 1000 orders are shown).

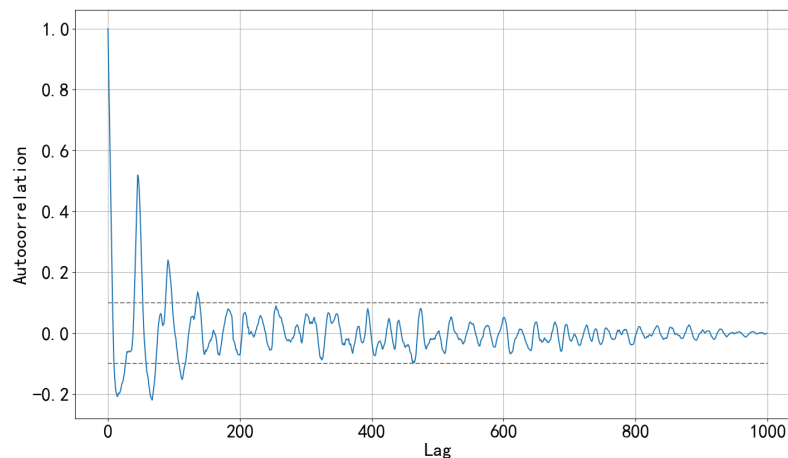


Figure 7. Sequence autocorrelation.

Therefore, we seek a time series model with long-term historical memory ability to decrease sequence autocorrelation to reduce residuals.

3.2. Time Series Model

LSTM is a commonly used model with long and short-term memory ability. Zhang et al. [28] adopted long short-term memory to reflect long-term memories of battery degradation tendency. Lu et al. [29] predicted electricity price based on LSTM. However, there is a risk of gradient vanishing and gradient explosion, and this is time-consuming due to the property of the back-propagation direction of time. A detailed discussion is shown in Appendix A. Hence, we use the TCN to avoid the disadvantages of LSTM and meet the needs better:

3.2.1. TCN Network

1. Dilated Causal Convolution

As shown in Figure 8, TCN has the characteristics of both causal convolution and dilated convolution [30]. Formally, for one-dimensional input sequence RS and a filter $f : 0, \dots, k-1 \rightarrow R$, the convolution result is,

$$F(t) = (RS *_d f)(t) = \sum_{i=0}^{k-1} f(i) \cdot RS_{t-d*i} \quad (6)$$

where k is the convolution filter size, d represents the sampling interval (expansion coefficient), and $t - d * i$ accounts for the direction of the past. When $d = 1$, it indicates that each point is collected, and the higher the level, the larger the d .

Causal convolution ensures that the output result at each time is determined by the current and previous time. The advantage of dilated convolution is that it increases the receptive fields without pooling loss information, so that each convolution output contains a large range of information. At the same time, the property of interval sampling avoids the problem of too many convolution layers and reduces the risk of gradient disappearance. Thus, it allows for both very deep networks and a very long effective history.

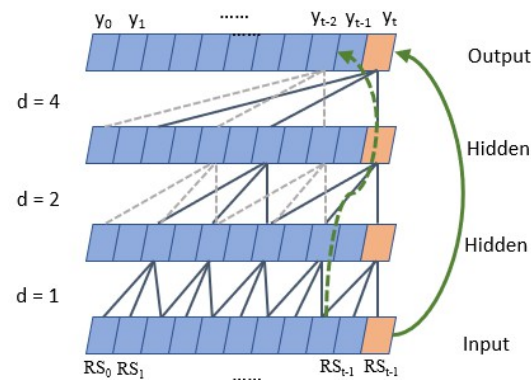


Figure 8. Dilated causal convolution. The green dashed and solid lines are residual connections. The blue and grey lines indicate dilated casual convolution method.

2. Residual connection

Residual connection transmits information directly between layers without convolution, which is a very effective way to train deep networks [31]. Figure 9 shows that in a residual block, the signal is transmitted by a weight layer composed of a series of operations such as causal expansion convolution and normalization, and also is directly mapped to the sum of the output of the weight layer,

$$o(t) = RS(t) + F(t) \quad (7)$$

where o is the output of the residual block. If $F(t) = 0$, the block turns into an identity map, and thus degenerates into a shallow network. This structure has the advantages of both the depth of the deep network and the shallow network. At the same time, even if the gradient of the weight layer is 0, the gradient of the entire residual network can be guaranteed not to be 0 due to the existence of the direct mapping mechanism. This block reduces the risk of vanishing gradients further when the model depth increases.

The advantages of the TCN can be analyzed as follows:

- In backpropagation, gradients do not accumulate along the time dimension, but along the network depth direction, and the number of network layers is often much smaller than the length of the input sequence. Furthermore, with the help of residual connection, the risk of gradient disappearance is further reduced, thus avoiding the gradient vanishing of RNN and LSTM networks.
- The TCN does not have to wait until the previous time is completed before calculating the result of the current time. Instead, all times are convoluted at the same time, which realizes parallelization and greatly reduces the running time.
- The dilated casual convolution mechanism of the TCN determines that it can use long-term historical information to predict by increasing the receptive field without dramatically increasing the model depth, which proves that it has the ability to remember long-term historical information.

In addition, in the copy memory task experiment [16], the TCN has no decline in memory ability in the sequence length from 0 to 250, and the 135 orders autocorrelation of our residual sequence is just in this range.

3. Formula of the TCN receptive field

$$R_{field} = 1 + 2 * (k - 1) * N_{stack} * \sum_i d_i \quad (8)$$

where N_{stack} represents the number of residual blocks in the TCN, and the sum of d_i indicates how many causal convolution layers are stacked in each residual block.

For example, $d = [1, 2, 3, 4]$ indicates that four causal convolution layers are stacked, and the expansion factors of each convolution layer are 1, 2, 3, and 4, respectively. We determine the parameters of the TCN according to the above formula and Figure 8, the sequence has the maximum 135-order autocorrelation, so the input sequence length is set to 150; since the first five orders have significant autocorrelation (the autocorrelation coefficients are all greater than 0.5), the convolution kernel size is set to 6; the length of the input sequence is not too large, so the number of residual blocks is set to 2. This way the receptive field is larger than the length of the input sequence. According to Formula (6), the depth is set to 4, $d = [1, 2, 4, 8]$. The maximum missing points interval is 24, so the output length of the whole model is set to 30.

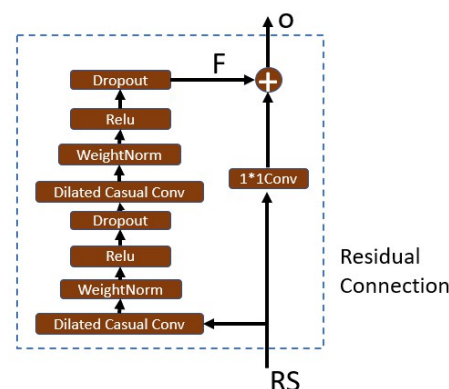


Figure 9. TCN residual block.

3.2.2. SE Attention Mechanism Module

SENet is a network model in computer vision, namely the squeeze-and-excitation network, which contains an attention module. It can learn the correlation between the output channels of the TCN module, assigns a weight to each feature channel, and pays more attention to those channels with key features, so as to improve the feature extraction ability of our method and the performance of the model. The SENet has a small amount of calculation and is very suitable for embedding into various networks. It mainly includes the steps of squeeze and excitation. See Appendix B for a detailed description.

4. The Overall Framework

The overall framework is shown in Figure 5. Firstly, we calibrate the installation matrix by the simulation data and actual data of different working conditions. Then we concatenate the residual errors between actual values and corrected simulation values in the installation coordinate system in chronological order for training the TCN-SE model. For the breakpoint t and the following p consecutive breakpoints, we correct the $1 + p$ simulation data under the installation coordinate system by the corrected installation matrix first, then add the corresponding residuals predicted by the trained model, and the total values are the complements for the breakpoint t and the following p breakpoints.

5. Experiment and Result Analysis

5.1. Continuous Sequence

5.1.1. Data Acquisition

For the residual sequence obtained in Section 2.4, select the point that meets the following requirement: 179 points after this point are uninterrupted. Among them, 150 points are used as model input, and 30 points are used as model output. Including this point, 180 sampling points are added to the continuous sequences. All such sequences are di-

vided into training sets and testing sets at a ratio of 8:2. For the sampling point n , such a continuous sequence form is,

$$\text{continuous_sequence_}n = [x_n, x_{n+1}, \dots, x_{n+149}, x_{n+150}, \dots, x_{n+179}] \quad (9)$$

5.1.2. Model Evaluation Metrics

To evaluate the performance of the model, we use the *MAE* and the error range. The *MAE* for continuous sequence n (abbreviated to *CSn*) is defined as,

$$MAE = \frac{\sum_{n=1}^N \sum_{t=150}^{179} |CSn_t - C\hat{S}n_t|}{30 * N} \quad (10)$$

where CSn_t and $C\hat{S}n_t$ represent the predicted value and the real value of the continuous sequence n at time t , respectively. N is the number of continuous sequences. The smaller the *MAE*, the better the performance of the model.

The error range is defined as,

$$\text{error range} = (\text{Minimum}(CSn_t - C\hat{S}n_t), \text{Maximum}(CSn_t - C\hat{S}n_t)) \\ t \in [150, 151, \dots, 179], n \in [1, 2, \dots, N]$$

5.1.3. Deep Learning Experiment

All the experiments were based on Windows operating system. Tensorflow (version 2.4.0) was selected as the deep learning framework. The GPU is NVIDIA RTX6000.

According to Formula (8), in order to make the receptive field cover the autocorrelation orders, we set the TCN-SE network parameters and training parameters as in Table 1:

Table 1. Model parameter settings.

Parameter	Settings
Number of residual blocks	2
Convolution kernel size	6
Dilations	[1, 2, 4, 8]
Model input length	150
Model output length	30
Number of iterations	100
Batch size	16
Loss function	MSE
Optimizer	Adam

5.1.4. Comparison of Different Models

The *MAE* and error range are used as evaluation indicators. The comparison of the prediction effects of LSTM, TCN, and TCN-SE models on the testing set is shown in Table 2. Compared with LSTM and TCN, the TCN-SE has significantly reduced *MAE*, error range, and time consumption.

Table 2. Model effect comparison.

Network	MAE (nT)	Error Range (nT)	Time Cost (s)
LSTM	112.39	(−446.24, 411.46)	545
TCN	67.27	(−225.98, 213.26)	176
TCN-SE	55.71	(−175.18, 168.28)	251

5.1.5. Prediction on Test Set

To compare with actual values, the predictions in the testing set are added by corrected simulation values in the installation coordinate system by the correction installation matrix from Section 2.4 of the corresponding time. Figures 10 and 11 show comparison of prediction effects on different days. The selected subfigures show residual errors of three models for the corresponding time. The subfigures show residual errors of three models in the selected time periods, which are always difficult to predict accurately. It can be seen that our method is accurate at almost every sampling point in the two figures, and has stronger long-term memory ability and more reasonable distribution mechanism.

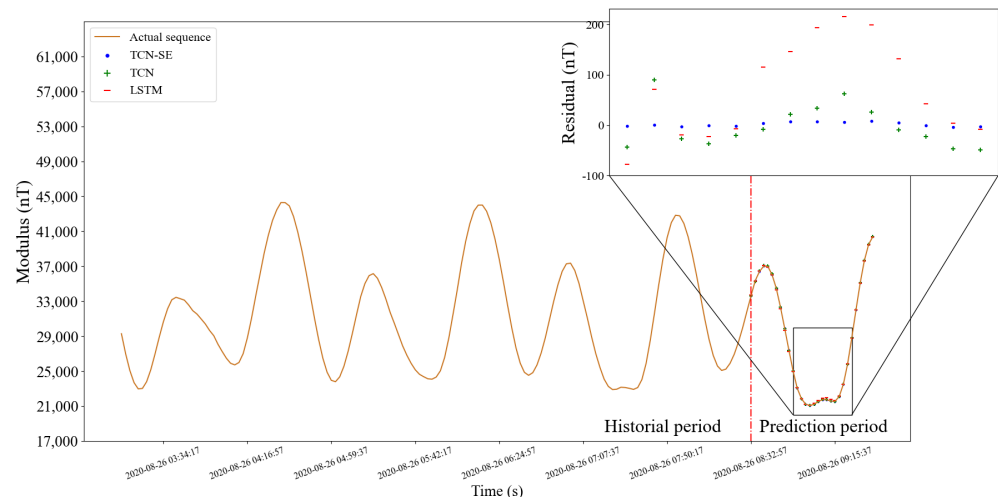


Figure 10. Prediction effect of three models. The orange curve is the actual sequence. The blue point, green plus symbol, and red horizontal line represent the predictions of the TCN-SE, TCN, and LSTM, respectively. We enlarged part of the forecast period to show the details of the residual errors of three models. Before and after the red dotted line are the historical and prediction periods, respectively.

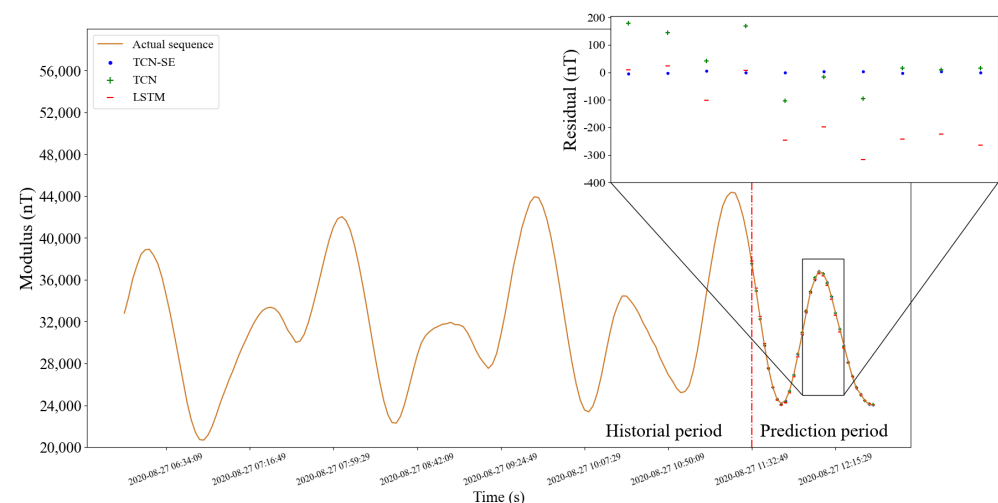


Figure 11. Prediction effect of the continuous sequence on different days from the above figure.

5.2. Sequence with Breakpoints

5.2.1. Data Acquisition

For the residual sequence obtained in Section 2.4, we select the breakpoint that meets the following requirement: the number of consecutive breakpoints after the breakpoint is less than 30. The first 150 sampling points before the breakpoint and the following 30 sampling points including the breakpoint are added to the sequences. For the breakpoint

m , it is assumed that p points are continuously missing after the breakpoint and expressed by *. The form of the sequence with breakpoint m (abbreviated to $SWBm$) is,

$$SWBm = [x_{m-150}, x_{m-149}, \dots, x_{m-1}, *, *, \dots, x_{m+p}, \dots, x_{m+29}] \quad (11)$$

5.2.2. Model Evaluation Metrics

The MAE for sequences with breakpoints is defined as,

$$MAE = \frac{\sum_{m=1}^M \sum_{t=p_m+150}^{179} |SWBm_t - \hat{SWBm}_t|}{\sum_{m=1}^n 30 - p_m}$$

where $SWBm_t$ and $\hat{SWBm}_t$ are, respectively, expressed as the real values and the model predicted values of breakpoint sequence m at time t . p_m is the number of breakpoints in sequence m . M is the number of sequences with breakpoints. The smaller the MAE, the better the performance of the model.

The error range is defined as,

$$\text{error range} = (\text{Minimum}(SWBm_t - \hat{SWBm}_t), \text{Maximum}(SWBm_t - \hat{SWBm}_t)) \\ t \in [p_m + 150, p_m + 151, \dots, 179], m \in [1, 2, \dots, M]$$

5.2.3. Comparison of Different Models

Since the missing points cannot be supplemented, we use the predictions after the breakpoints in the forecast period, which are at the time without missing values to demonstrate the extrapolation and the complementation ability of the method. In addition, there is a little difference from before. The predicted residuals at the time of missing points are added to the actual residual sequence as historical residual values, to avoid the situation that the number of continuous historical data before some breakpoints is less than the model input length. At the same time, we realize rolling complement and prediction. The MAE and error range in Section 5.2.2 are used as evaluation indicators. The comparison of the prediction effects of LSTM, TCN, and TCN-SE models which are trained on continuous sequences on the test set is shown in Table 3. We found a similar conclusion to that in Section 5.1.4.

Table 3. Model comparison.

Network	MAE (nT)	Error Range (nT)
LSTM	144.15	(−483.42, 442.98)
TCN	92.19	(−254.67, 259.26)
TCN-SE	74.63	(−215.18, 211.63)

5.2.4. Prediction Effect

Similar to Figure 10, the orange curve in Figures 12 and 13 is the actual values. The blue breakpoints consist of the sum of 30 (model output length) predictions after breakpoints and corrected simulation values in the installation coordinate system of corresponding time. The subfigures show the residual errors in the corresponding periods.

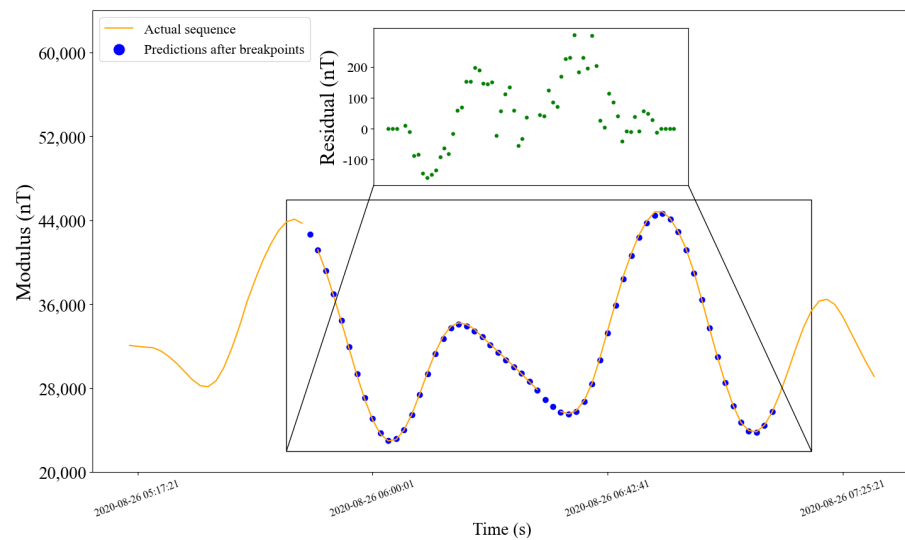


Figure 12. Prediction effect of the sequence with breakpoints. The orange curve is the actual data. Dark blue points are predictions after breakpoints to highlight locations. The subfigure shows the residual errors of the two curves in the selected period.

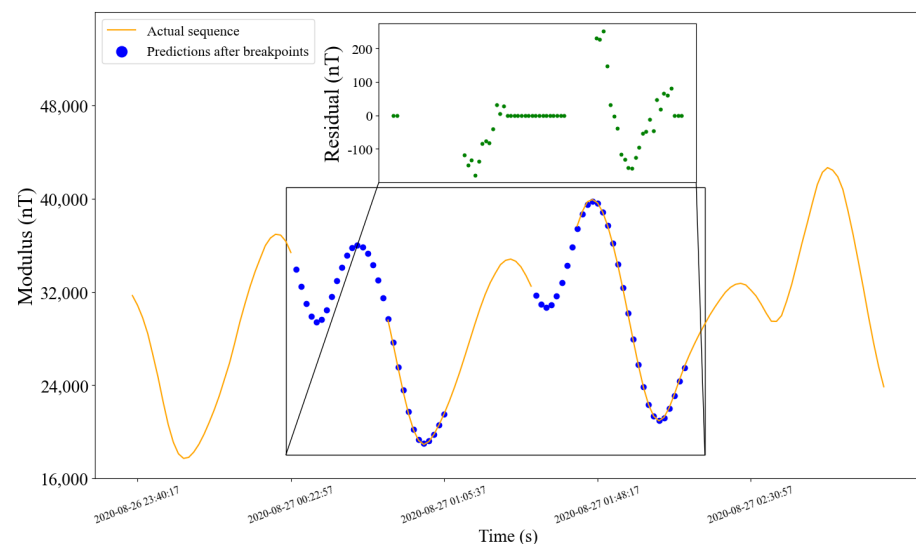


Figure 13. Prediction effect of the sequence with breakpoints on different days from the above figure.

6. Conclusions

In this paper, a two-step model is proposed, which first corrects the installation matrix under different working conditions and uses the deep learning method to fit the residual errors. Considering the different effects of the external field, lithosphere, and stray magnetic field in different directions, the installation matrix is corrected separately in two working conditions. Because the residual sequence contains autocorrelation and nonlinear information, the TCN network with dilated causal convolution mechanism is used for fitting, and the SE model with attention mechanism is used to improve the accuracy. Experiments show that the TCN-SE used in this paper has an MSE of 74.63 nT and an error range of $[-215.18 \text{ nT}, 211.63 \text{ nT}]$ when predicting the subsequent data of missing values. The extrapolation ability of our method is proved, and it can be used to complement the missing values of on-orbit magnetic data.

However, the installation matrix is not only affected by two working conditions of directing the Sun and the Earth, but may also be related to the disturbance caused by different tasks (working conditions). In order to get more detailed classifications of working

conditions and further reduce the linear error of the installation matrix, we consider clustering the residual sequence between the actual values and the inertial coordinate system values after being multiplied by the installation matrix while directing the Sun and the Earth time, respectively.

The universal magnetic map updates every five years, its magnetic isoplane moves 10–15 km every year, and the accumulated error will exceed 100 km in 10 years [32]. We consider that the special magnetic model established in this paper can also replace the magnetic map-matching method. It updates regularly according to the updated telemetry data, and can be used for magnetic navigation and satellite attitude determination after the extrapolated data are interpolated to meet the actual frequency requirements.

Author Contributions: Conceptualization, W.C. and R.Z.; Formal analysis, C.S.; Investigation, W.C.; Methodology, X.L.; Resources, R.Z., Y.Z. and X.L.; Supervision, R.Z.; Validation, W.C.; Writing—original draft, W.C.; Writing—review & editing, W.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A. Discussion of LSTM Disadvantages

LSTM is a commonly used model with long and short memory, but it has the following two disadvantages:

1. LSTM cannot completely eliminate the risk of gradient descent in a long sequence and may cause gradient explosion.

We use the simplified version of the LSTM block diagram shown in Figure A1 to illustrate this problem. For the parameter θ , the gradient formula for reverse conduction is:

$$\frac{\partial \varepsilon}{\partial \theta} = \frac{\partial \varepsilon}{\partial c_n} \sum_{k=1}^n \left(\left(\prod_{i=k+1}^n \frac{\partial c_i}{\partial c_{i-1}} \right) \frac{\partial c_k}{\partial \theta} \right) \quad (\text{A1})$$

$$\frac{\partial c_i}{\partial c_{i-1}} = \frac{\partial (f_i \odot c_{i-1} + I_i \odot \tilde{c}_i)}{\partial c_{i-1}} = \text{diag}(f_i) + \frac{\partial (I_i \odot \tilde{c}_i)}{\partial c_{i-1}} \quad (\text{A2})$$

where ε is the difference between the output of the output gate and the actual value, $[x_1 x_2 \dots x_n]$ is the input sequence; I, f, c, h and o represent the state of the cell state, forget gate, hidden layer, and output gate. If the second term on the right side of the above equation can be ignored, since the value range of the forget gate state is from 0 to 1, there is still a risk of gradient disappearance. However the first term is not multiplied by the weight matrix, thus the term will not be reduced by the weight or the bias parameter, and the property of the tanh function determines that the forget gate will always have some mappings to -1 and 1 , so the risk of gradient disappearance is much smaller than that of an RNN. If the second term on the right side of the above equation is not negligible, the sum with the first term may be greater than 1, so there is a risk of gradient explosion.

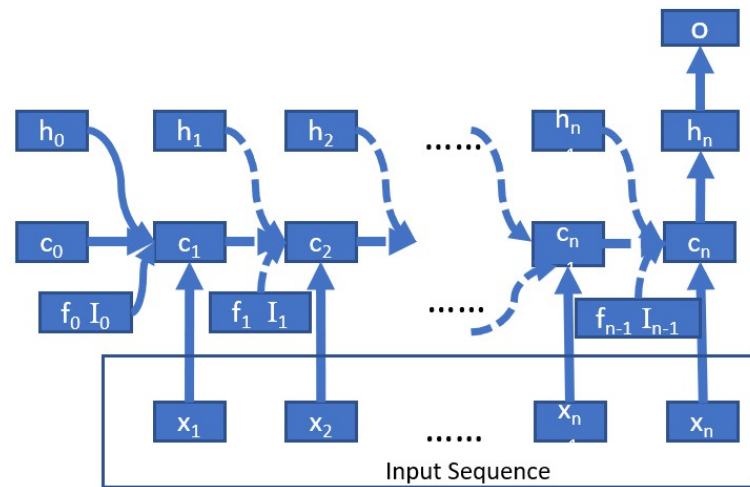


Figure A1. LSTM simplified structure.

- Each LSTM unit has four full connection layers. If the time span is long, the network will be deep, and calculations will be cumbersome and time-consuming.

In addition, Ref. [33] proved that LSTM has geometric ergodicity and, therefore, does not have long-term memory ability. From the comparison results of the copy memory task experiment [16], its long-term memory ability gradually decreases with the increase of sequence length.

Appendix B. SE attention mechanism

Appendix B.1. Squeeze and Excitation

The squeeze function is defined as:

$$z_c = F_{sq}(U) = \frac{1}{H * W} \sum_{i=1}^H \sum_{j=1}^W u_c(i, j) \quad (A3)$$

where z_c denotes the c -th element of squeezed channels, $F_{sq}(\cdot)$ denotes the squeeze function, $U = [u_1, u_2 \dots u_n]$ denotes the whole 3-D features from tcn network whose shape is (H, W, C) and $u_c(i, j)$ denotes the data at row i column j of u_c . After squeezing the input features U by function $F_{sq}(\cdot)$, a vector is generated with z_c be the c -th element of z whose shape is $(1, 1, C)$.

Let the matrix $W1 \in C \times C/r$ and $W2 \in C/r \times C$ be two fully connected layers with bottleneck structure whose compression ratio is r . Let $\sigma1(\cdot)$ as well as $\sigma2(\cdot)$ denote the reLU function and the sigmoid function, respectively. The excitation function F_{ex} is defined as:

$$s = F_{ex}(z) = \sigma2(W2\sigma1(W1z)) \quad (A4)$$

After the squeeze and excitation steps, we get an attention vector s , which we can use to adjust the channels of U . The formula is defined as follows:

$$y_c = F_{scale}(s_c, u_c) = s_c * u_c \quad (A5)$$

where $Y = [y_1, y_2 \dots y_n]$ denotes the final output of the squeeze-and-excitation process.

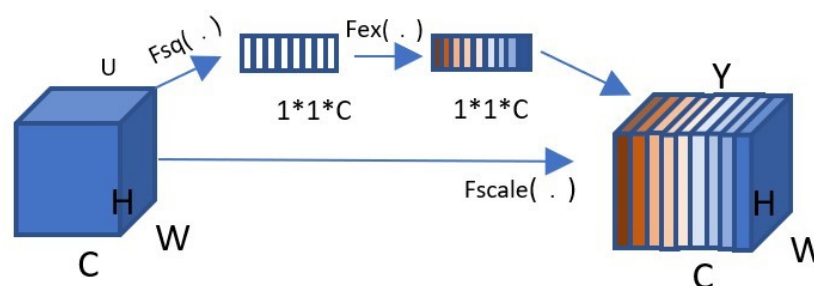


Figure A2. SENet structure.

References

- Gao, D.; Zhang, T.; Cui, F.; Li, M. Development of Geomagnetic Navigation and Study of High Precision Geomagnetic Navigation for LEO Satellites. In Proceedings of the Second China Aerospace Safety Conference, Brussels, Belgium, 14–16 June 2017; pp. 438–445.
- Macmillan, S.; Finlay, C. The international geomagnetic reference field. In *Geomagnetic Observations and Models*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 265–276.
- Lippi, M.; Bertini, M.; Frasconi, P. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 871–882. [\[CrossRef\]](#)
- Zhu, M. A Study on SVM Algorithm for Missing Data Processing. Master's Thesis, Tianjin University, Tianjin, China, 2017.
- Gao, W.; Niu, K.; Cui, J.; Gao, Q. In A data prediction algorithm based on BP neural network in telecom industry. In Proceedings of the 2011 International Conference on Computer Science and Service System (CSSS), Nanjing, China, 27–29 June 2011; pp. 9–12.
- Li, L.; Li, Y.; Li, Z. Efficient missing data imputing for traffic flow by considering temporal and spatial dependence. *Transp. Res. Part Emerg. Technol.* **2013**, *34*, 108–120. [\[CrossRef\]](#)
- Alsharif, M.H.; Younes, M.K.; Kim, J. Time series ARIMA model for prediction of daily and monthly average global solar radiation: The case study of Seoul, South Korea. *Symmetry* **2019**, *11*, 240. [\[CrossRef\]](#)
- Rahman, M.A.; Yunsheng, L.; Sultana, N. Analysis and prediction of rainfall trends over Bangladesh using Mann–Kendall, Spearman's rho tests and ARIMA model. *Meteorol. Atmos. Phys.* **2017**, *129*, 409–424. [\[CrossRef\]](#)
- Wu, X.; Wang, Y. Extended and Unscented Kalman filtering based feedforward neural networks for time series prediction. *Appl. Math. Model.* **2012**, *36*, 1123–1131. [\[CrossRef\]](#)
- Xiu, C.; Ren, X.; Li, Y.; Liu, M. Short-Term Prediction Method of Wind Speed Series Based on Kalman Filtering Fusion. *Trans. China Electrotech. Soc.* **2014**, *29*, 253–259.
- Zeng, J.; Qiao, W. Short-term solar power prediction using a support vector machine. *Renew. Energy* **2013**, *52*, 118–127. [\[CrossRef\]](#)
- De Brabanter, K.; De Brabanter, J.; Suykens, J.A.; De Moor, B. Approximate confidence and prediction intervals for least squares support vector regression. *IEEE Trans. Neural Netw.* **2010**, *22*, 110–120. [\[CrossRef\]](#)
- Tokgoz, A.; Unal, G. In A RNN based time series approach for forecasting turkish electricity load. In Proceedings of the 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2–5 May 2018; pp. 1–4.
- Hundman, K.; Constantinou, V.; Laporte, C.; Colwell, I.; Soderstrom, T. In Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 387–395.
- Jian, X.; Gu, H.; Wang, R. A short-term photovoltaic power prediction method based on dual-channel CNN and LSTM. *Electr. Power Sci. Eng.* **2019**, *35*, 7–11.
- Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
- Oord, A.v.d.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv* **2016**, arXiv:1609.03499.
- Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. In Language modeling with gated convolutional networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 933–941.
- Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. In Convolutional sequence to sequence learning. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1243–1252.
- Shi, C.; Zhang, R.; Yu, Y.; Sun, X.; Lin, X. A SLIC-DBSCAN Based Algorithm for Extracting Effective Sky Region from a Single Star Image. *Sensors* **2021**, *21*, 5786. [\[CrossRef\]](#) [\[PubMed\]](#)
- Olsen, N.; Albin, G.; Bouffard, J.; Parrinello, T.; Tøffner-Clausen, L.J.E. Magnetic observations from CryoSat-2: Calibration and processing of satellite platform magnetometer data. *Earth Planets Space* **2020**, *72*, 1–18. [\[CrossRef\]](#)
- Liu, Y.; Wang, S.; Zhang, J.; Zheng, Y.; Qiao, Y. Research on the eleventh generation IGRF. *Acta Seismol. Sin.* **2017**, *35*, 125–134.
- Yan, F.; Zhen-Chang, A.; Han, S.; Fei, M.J. Analysis of variation in geomagnetic field of Chinese mainland based on comprehensive model CM4. *Acta Phys. Sin.* **2010**, *59*, 8941–8953.
- Wang, D.J. Analysis of the international geomagnetic reference field error in the China continent. *Chin. J. Geophys.* **2003**, 171–174.

25. Chen, B.; Gu, Z.W.; Gao, J.T.; Yuan, J.H.; Di, C.Z. Geomagnetic secular variation in China during 2005–2010 described by IGRF-11 and its error analysis. *Prog. Geophys.* **2012**, *27*, 512–521.
26. Pang, H.; Chen, D.; Pan, M.; Luo, S.; Zhang, Q.; Luo, F.J. Nonlinear temperature compensation of fluxgate magnetometers with a least-squares support vector machine. *Meas. Sci. Technol.* **2012**, *23*, 025008. [[CrossRef](#)]
27. Shorshi, G.; Bar-Itzhack, I.Y. Satellite autonomous navigation based on magnetic field measurements. *J. Guid. Control. Dyn.* **1995**, *18*, 843–850. [[CrossRef](#)]
28. Zhang, Y.; Xiong, R.; He, H.; Pecht, M. Long short-term memory recurrent neural network for remaining useful life prediction of lithiumion batteries. *IEEE Trans. Veh. Technol.* **2018**, *67*, 5695–5705. [[CrossRef](#)]
29. Peng, L.; Liu, S.; Liu, R.; Wang, L. Effective long short-term memory with differential evolution algorithm for electricity price prediction. *Energy* **2018**, *162*, 1301–1314. [[CrossRef](#)]
30. Lea, C.; Flynn, M.D.; Vidal, R.; Reiter, A.; Hager, G.D. In Temporal convolutional networks for action segmentation and detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 156–165.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. In Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
32. Komolkin, A.V.; Kupriyanov, P.; Chudin, A.; Bojarinova, J.; Kavokin, K.; Chernetsov, N.J. Theoretically possible spatial accuracy of geomagnetic maps used by migrating animals. *J. R. Soc. Interface* **2017**, *14*, 20161002. [[CrossRef](#)] [[PubMed](#)]
33. Zhao, J.; Huang, F.; Lv, J.; Duan, Y.; Qin, Z.; Li, G. Do rnn and lstm have long memory? In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 11365–11375.