International Journal of
Environmental Research and Public Health

MDPI

Article

BiLSTM-I: A Deep Learning-Based Long Interval Gap-Filling Method for Meteorological Observation Data

Chuanjie Xie [1,†], Chong Huang [1,*,†], Deqiang Zhang [2] and Wei He [1]

[1] State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China; xiecj@lreis.ac.cn (C.X.); hew.20s@igsnrr.ac.cn (W.H.)
[2] Key Laboratory of Plant Resources Conservation and Sustainable Utilization, South China Botanical Garden, Chinese Academy of Sciences, Guangzhou 510650, China; zhangdeq@scbg.ac.cn
* Correspondence: huangch@lreis.ac.cn
† These authors contributed equally to this work.

Abstract: Complete and high-resolution temperature observation data are important input parameters for agrometeorological disaster monitoring and ecosystem modelling. Due to the limitation of field meteorological observation conditions, observation data are commonly missing, and an appropriate data imputation method is necessary in meteorological data applications. In this paper, we focus on filling long gaps in meteorological observation data at field sites. A deep learning-based model, BiLSTM-I, is proposed to impute missing half-hourly temperature observations with high accuracy by considering temperature observations obtained manually at a low frequency. An encoder-decoder structure is adopted by BiLSTM-I, which is conducive to fully learning the potential distribution pattern of data. In addition, the BiLSTM-I model error function incorporates the difference between the final estimates and true observations. Therefore, the error function evaluates the imputation results more directly, and the model convergence error and the imputation accuracy are directly related, thus ensuring that the imputation error can be minimized at the time the model converges. The experimental analysis results show that the BiLSTM-I model designed in this paper is superior to other methods. For a test set with a time interval gap of 30 days, or a time interval gap of 60 days, the root mean square errors (RMSEs) remain stable, indicating the model's excellent generalization ability for different missing value gaps. Although the model is only applied to temperature data imputation in this study, it also has the potential to be applied to other meteorological dataset-filling scenarios.

Keywords: time series; data imputation; deep learning; meteorological observation data

Citation: Xie, C.; Huang, C.; Zhang, D.; He, W. BiLSTM-I: A Deep Learning-Based Long Interval Gap-Filling Method for Meteorological Observation Data. *Int. J. Environ. Res. Public Health* **2021**, *18*, 10321. https://doi.org/10.3390/ijerph181910321

Academic Editor: Paul B. Tchounwou

Received: 29 August 2021
Accepted: 23 September 2021
Published: 30 September 2021

1. Introduction

Temperature is a very important variable for agricultural and ecosystem studies, and it is an essential input in agricultural crop growth simulations, agrometeorological disaster monitoring, and ecosystem simulations [1,2]. As agricultural and ecological simulations have improved, the resolution requirements for temperature data have increased; notably, high-resolution data are needed in wind monitoring in dry and hot areas, agrometeorological hazard assessments, and simulations of carbon emissions from forest block ecosystems [3,4]. Temperature observations are usually obtained from field meteorological stations, and the data observed at small weather stations commonly have gaps due to equipment failure, harsh environmental conditions or operational errors [5]. The imputation or completion of missing data is an essential preprocessing task before temperature observation data are applied.

There are various methods for data imputation, and they can be classified into three main categories: deterministic model-based methods, statistical model-based methods and machine learning methods [6]. Deterministic models are based on observed values and can

interpolate missing values using deterministic mathematical methods, such as the overall average, nearest neighbour, polynomial, and spline function interpolation methods for unobserved values [7,8]. Imputation methods based on statistical models fully consider the observation error and the error generated in the imputation process to reduce the error in the imputation results through optimization. The regression method is representative of such methods, and it obtains mathematical expressions of observed values through regression and then interpolates the missing values using mathematical expressions [9]; various time series regression imputation methods have been widely used [10,11]. An imputation method that combined a Kalman filter and time series regression analysis performed well in the imputation of missing values in single-factor time series [5,12]. The accuracy of time series data imputation depends on the closeness between the time series representation model in the algorithm and the 'real' model. Traditional methods of time series data imputation generally assume a predefined model structure for time series data. The fitness of the predefined model has a great impact on the accuracy of data imputation.

Alternatively, machine learning involves learning the potential distribution of data from the acquired observations and interpolating missing values with a model established after learning. Data imputation methods based on traditional machine learning include those based on principal component analysis, low-rank matrix decomposition, kernel methods [13–15], and combined data imputation methods [16]. Modern machine learning imputation methods can be applied in data imputation by applying deep learning techniques; this approach provides a rich and diverse network structure [17,18] and is suitable for univariate or multivariate time-series imputation [19,20]. The active learning process may obtain a better representation model much closer to the real data structure, thus obtaining a higher data imputation accuracy.

Meteorological observations are typical time-series data. Time-series imputation methods, such as mean imputation, stochastic regression imputation are generally available for filling in missing values in meteorological observations. Although methods of imputing missing values in time series are abundant, research on how to use low-frequency manually acquired observations to fill the long time interval gaps in high-frequency machine-based observations is lacking [21]. Considering a common situation, an ecological station collects the temperature data using an automatic weather station in the field, and manual temperature observation is also employed at the same time. The temporal frequency of the observation product of the automatic meteorological data output is high, for example, one record per 30 min, with 48 observation records per day; manual temperature observations are obtained in the morning, at midday and in the evening three times per day, resulting in only three manual observation records. Although the automatic meteorological data are superior to manual observations on recording frequency, they are greatly affected by occasional factors, such as the bad weather, the problem of facilities, etc., which might easily lead to long-time-interval data loss.

In this study, we proposed a new deep learning-based model BiLSTM-I to obtain complete half-hourly-frequency temperature observation datasets based on daily manually observed temperature data. We detailed the model structure. Taking a forest ecology station in Guangzhou, China as an example, we elaborated the application of our model to fill the long time interval gap of automatic temperature observation data. Moreover, we compared our results with other classical methods for missing data imputation to highlight the efficiency of our model.

2. Materials and Methods

2.1. Meteorological Temperature Observation Data

In this study, meteorological temperature observation data from the National Field Scientific Observation and Research Station of the Dinghushan Forest Ecosystem (23.18° N, 112.53° E) in Guangzhou, China, were used. This ecosystem observatory has performed comparative temperature observation experiments with both manual observations and automatic meteorological machine-based observations and has a long record of temperature

observation data. The following Table 1 lists the manual temperature observation data and automatic machine observation data used in this study.

Table 1. Information table of temperature observation data set.

Dataset	Frequency	Time	Missing Values
Dataset 1	8 am daily	2018/11/13–2020/2/10	None
Dataset 2	2 pm daily	2018/11/13–2020/2/10	None
Dataset 3	8 pm daily	2018/11/13–2020/2/10	None
Dataset 4	Every half hour	2018/11/13–2020/2/10	Short time interval gaps and one long time interval gap

(Datasets 1–3 include manual observation data, and Dataset 4 includes automatic observation data).

Since the Dinghushan Ecological Station is located in the mountainous region of southern China, the automatic observation equipment is susceptible to external effects, which may lead to missing observation records for long periods of time, especially in the thunderstorm season. Figure 1 shows the distribution of missing meteorological observation data; notably, there were missing temperature observations for more than 2 months around July 2020.

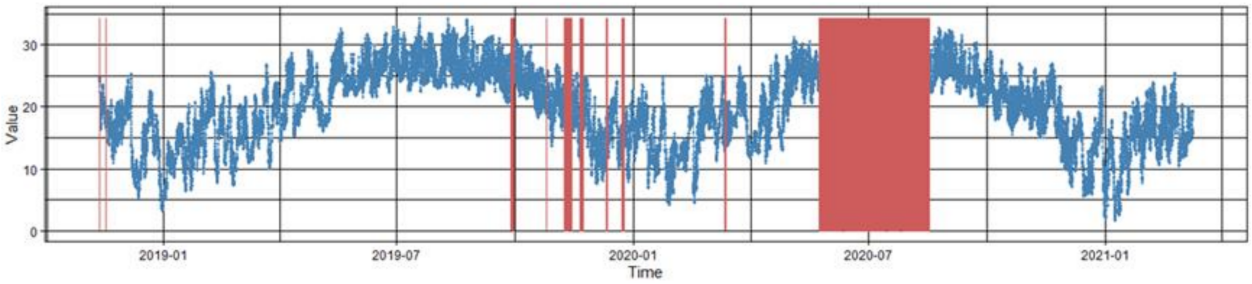

Figure 1. Distribution of missing values in half-hourly temperature observation data.

This article focuses on the imputation of missing machine temperature observations for more than 2 months around July 2020 using manual observations obtained three times a day. Since linear correlations can be easily established between manual and machine observations, the core objective of the data imputation problem is determining how to apply low-frequency manually obtained temperature observations to fill long-time-interval gaps in data sets of high-frequency automatic machine temperature observations.

2.2. Baseline Methods

2.2.1. Time Series Data Imputation with Kalman Smoothing

Kalman smoothing has the same mathematical basis as the widely used Kalman filter, both of which involve estimating unobservable system states from observable data. The Kalman filter method has linear and non-linear forms, and the basic linear Kalman filter equation is used in this case. The evolution of the system state space can be expressed as Equations (1) and (2) [22,23].

$$a_t = T_t a_{t-1} + R_t \eta_t \qquad \eta_t \sim N(0, Q_t) \tag{1}$$

$$y_t = Z_t a_t + \varepsilon_t \qquad \varepsilon_t \sim N(0, H_t) \tag{2}$$

where a_t is an unobservable system state, T_t is the state transfer matrix, R_t is the system noise-driven matrix, y_t is the observed data, and Z_t is the observation matrix. η_t and ε_t denote the white noise of the state transform process and measurement, and they are independent of each other.

Following the Kalman filter and smoothing methods, the best estimate of the system state $\widetilde{a}_{t|n}$ can be obtained assuming an observation set $Y_t = (y_1, y_2, \cdots, y_n)$ with n samples; the corresponding estimation error covariance matrix is $P_{t|n} = \left(a_t - \widetilde{a}_{t|n}\right)^T \left(a_t - \widetilde{a}_{t|n}\right)$.

Kalman filtering provides an estimation of the current system state from observations, and smoothing yields an estimation of the past system state; the best estimation processes for specific system states have been described in many studies [24].

Kalman smoothing, which uses all temperature observations available before and after the missing value window, provides the best estimation of the state at any moment in a previous observation period and can be used to obtain valid estimates of the missing temperature observations. To apply Kalman smoothing, a state space model, such as that in Equations (1) and (2), is required. These equations include the matrices T_t, Z_t, and R_t. The state equations are developed using a structured time series model and a time series regression model.

(1) Structured time series model (Kalman-S)

The basic structured (BSM: basic structured model) time series model is used here, and the basic BSM formulas are as follows Equations (4)–(6) [25,26]:

$$y_t = \mu_t + \gamma_t + \varepsilon_t \tag{3}$$

$$\mu_t = \mu_{t-1} + \beta_t + \eta_t \tag{4}$$

$$\beta_t = \beta_{t-1} + \xi_t \tag{5}$$

$$\gamma_t = -\sum_{j=1}^{s-1} \gamma_{t-j} + \omega_t \tag{6}$$

In the above equation set, Equation (3) is the observed equation for time series y_t, where μ_t is the trend component and is linearly approximated by Equations (4) and (5); γ_t is the seasonal component of the time series, which is defined by Equation (6); ε_t, η_t, ξ_t and ω_t in the above equations are the mean zero and variance of δ_ε^2, δ_η^2, δ_ξ^2 and δ_ω^2 for mutually independent noise, respectively; s in Equation (6) is the number of seasonal cycles of the time series in a year.

By transformation, the BSM equations can be transformed into state model expression form. For simplicity, we can set s to 4 and obtain Equations (7) and (8):

$$a_t \equiv \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \\ \gamma_{t-2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mu_{t-1} \\ \beta_{t-1} \\ \gamma_{t-1} \\ \gamma_{t-2} \\ \gamma_{t-3} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \eta_t \\ \xi_t \\ \omega_t \end{bmatrix} \tag{7}$$

$$y_t = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \end{bmatrix} a_t + \varepsilon_t \tag{8}$$

By comparing Equations (7) and (8) with the Kalman smoothing state Equations (1) and (2) above, the expressions required to transform the BSM equations into a state model can be obtained.

(2) ARIMA-based state space model (Kalman-A)

The differential integrated moving average autoregressive model (ARIMA: autoregressive integrated moving average) is a widely used time-series forecasting method and is also widely used in single-factor time-series analysis [27]. The ARIMA-based state model has been applied to problems involving traffic state forecasting and missing value imputation for time series [5,28]. Compared with ARMA (autoregressive moving average model), ARIMA first enhances the stability of observed time series through difference operations, and ARMA is then used to model the time series. The mathematical expressions of both are consistent, and ARIMA is used below in the introduction of the state model establishment process [5,29,30].

2.2.2. BRITS-I Time Series Imputation Method Based on Deep Learning

Deep learning is an effective method for the imputation of time series data [31], for example, a recurrent neural network (RNN) was used to impute missing values in a smooth fashion [10]. The BRITS-I method [32] uses RNN to predict the missing values directly in a recurrent dynamical system based on the observed data. As a simpler case of BRITS-I, RITS-I employs a unidirectional recurrent dynamical system, in which the missing value in the time series can be derived by its predecessors with a fixed arbitrary function. The algorithm contained a recurrent component implemented by a RNN and a regression component represented by a fully-connected network. A standard recurrent network [17] can be represented as Equation (9):

$$h_t = \sigma(W_h h_{t-1} + U_h x_t + b_h) \quad (9)$$

where σ is the sigmoid function, W_h, U_h and b_h are parameters, and h_t is the hidden state of previous time steps.

Considering that the time series may be irregularly sampled, a temporal decay factor γ_t was introduced in RITS-I, which represents the missing patterns in the time series Equation (10).

$$\gamma_t = exp\{-max(0, W_\gamma \delta_t + b_\gamma)\} \quad (10)$$

In a unidirectional recurrent dynamical system, errors of estimated missing values are delayed until the presence of the next observation. To alleviate the issue, BRITS-I utilized the bidirectional recurrent dynamics on the given time series, i.e., besides the forward direction, each value in time series can be also derived from the backward direction by another fixed arbitrary function [32].

2.3. BiLSTM-I Model Development

Several studies showed that neural networks with sequence-to-sequence (Seq2Seq) structures can efficiently fill gaps in time series [32,33]. However, deep learning models, with different structures, designs and optimization objective functions, can exhibit large performance differences when solving similar problems. The imputation model BiLSTM-I proposed in this paper designed an encoder-decoder deep learning architecture, and an optimization objective error function, to obtain high accuracy in long interval gap filling for time-series meteorological observation data.

2.3.1. Basic Definition

The temperature displays periodicity on the scale of days, and it is natural to divide long time series of half-hourly temperature observations over days into a segmented series of 48 observations per day. To focus on the imputation of missing values over long time intervals, occasional or short-term gaps in the time series are first interpolated using the Kalman smoothing method described above. The temperature time series thus included two segment types: daily segments without missing values, denoted as d^j_{full}, and daily segments containing observations in the morning, afternoon, and evening, denoted as d^j_{miss}. The time-segmented series can be expressed as Equation (11):

$$\left\{ d^1_{full}, \ldots, d^i_{full}, d^{i+1}_{miss}, \ldots, d^{i+m}_{miss}, d^{i+m+1}_{full}, \ldots, d^n_{full} \right\} \quad (11)$$

In this study, we used temperature time series of two years. Therefore, sequence (11) is a temperature time series of length 730 days (n). The missing value window width of m was set to 30 and 60 days, respectively. For the half-hourly temperature observation, sequence (11) represents a temperature observation data sequence of length 35,040 (L) with 1440 and 2880 missing values expressed in the form of daily segmentation.

To represent the positions of the missing values in the time series (11), a mask time series $\{m_t^i\}$ of corresponding length L is constructed using Equation (12) for the half-hourly sampled temperature time series $\{T_t^i\}$ of length L, where:

$$m_t^i = \begin{cases} 0\ , As\ T_t^i\ Unobserved \\ 1\quad , \qquad\qquad else \end{cases} \tag{12}$$

Now, the half-hourly mask sequence of length L is segmented in days, the mask without missing values is segmented daily as M_{full}^j, and the mask containing only three observations in the morning, afternoon and evening is segmented daily as M_{miss}^j; therefore, the mask sequence corresponding to (11) segmented in days can be constructed as (13):

$$\left\{ M_{full}^1, \ldots,\ M_{full}^i,\ M_{miss}^{i+1}, \ldots,\ M_{miss}^{i+m},\ M_{full}^{i+m+1}, \ldots,\ M_{full}^n \right\} \tag{13}$$

2.3.2. Rolling Window Sampling

A rolling window approach is used to construct a sample set for deep learning model training based on the time series segmented by day. For the time interval gap length of m days, the length of the rolling window needs to be constructed to be greater than m, and observations of length s (days) are kept at each end of m so that the rolling window length w is $m + 2 \times s$ days. The training samples are constructed by adapting the Seq2Seq training method to the training input sample of length w; the temperature observation sequence (14) is:

$$\{\ d_{full}^1, \ldots,\ d_{full}^s,\ d_{miss}^{s+1}, \ldots,\ d_{miss}^{s+m},\ d_{full}^{s+m+1}, \ldots,\ d_{full}^w \} \tag{14}$$

The following time series output (15) can be obtained from the training process:

$$\{\ d_{full}^1, \ldots,\ d_{full}^s,\ \hat{d}_{impt}^{s+1}, \ldots,\ \hat{d}_{impt}^{s+m},\ d_{full}^{s+m+1}, \ldots,\ d_{full}^w \} \tag{15}$$

In sequence (15), $\hat{d}_{impt}^j$ is the complete segment of the observed temperature values for each half-hour in a day after the imputation of missing values. In constructing the training samples, a mask sequence of length w (days) is constructed as another input to the training samples according to the observation values corresponding to the mask sequence on the order of days.

The training sample is constructed based on the temperature observation sequence without missing values, and the pattern of missing observations in the sample is consistent with the actual situation; only three observations per day in the morning, afternoon, and evening are considered. Table 2 gives an example of a day of temperature data with missing values in the training sample and the corresponding mask.

2.3.3. Design of Deep Learning Models

Typical Seq2Seq-based deep learning models for the imputation of time series data are SSIM and BRITS-I [34,35]. In this paper, the advantages of these models are utilized, an encoder-decoder deep learning architecture is adopted, and the structure of the designed deep learning model is shown in the following figure. For convenience, the above input sequence (13) is denoted as $x = \{x_1, x_2..., x_n\}$, the output sequence (14) is denoted as $y = \{y_1, y_2..., y_n\}$, and the mask sequence (12) is denoted as $m = \{m_1, m_2..., m_n\}$.

(1) Encoder

As shown in Figure 2, the basic structure of the encoding part of the deep learning structure is based on LSTM-I. The core of the LSTM-I unit is an LSTM neural network unit. The recurrent neural network unit directly adopts a long short-term memory (LSTM) unit, which is a special kind of RNN, to solve the gradient disappearance and gradient explosion problems during the training of long sequences. In addition, the missing value part of the

temperature observation set in this paper, with 48 half-hourly temperature values daily, contains only 3 observations, so the variable y_t in Equation (10) for missing value intervals is not used.

Table 2. Example of data for a day within the window of missing values in the sample series.

Time	Temperature Value	Mask
1	Na	0
2	Na	0
...	...	...
17	17.14	1
18	Na	0
...	...	...
27	Na	0
28	21.78	1
29	Na	0
...	...	...
39	Na	0
40	19.86	1
41	Na	0
...	...	...
47	Na	0
48	Na	0

The first column of the table is the time sequence number in half hours, starting from 0:00, and the middle column is the temperature value (in Celsius). Only three valid observations are available (morning, midday and evening); the third column is the missing position mask for temperature observations.

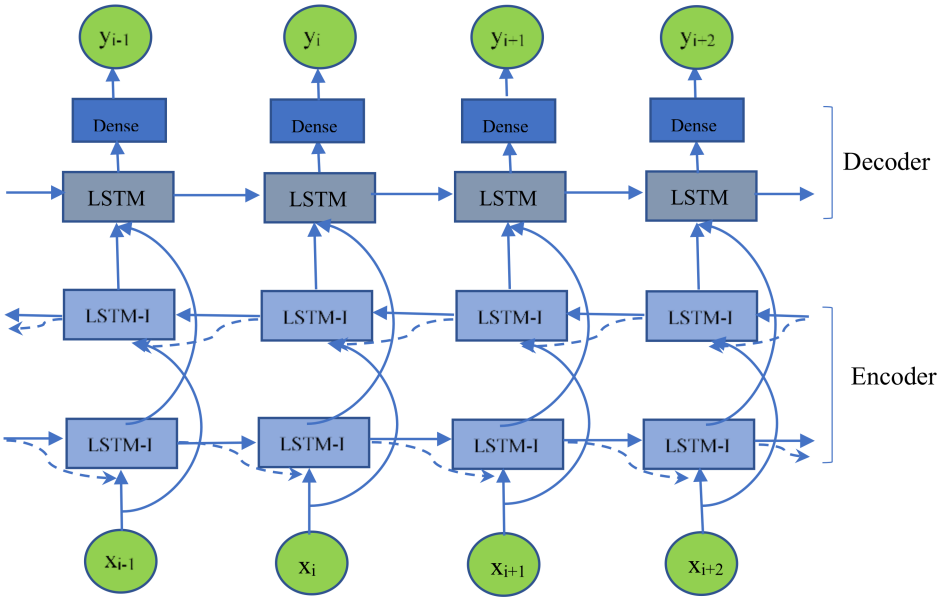

Figure 2. Structure of the imputation neural network for missing temperature values.

The LSTM structure and mathematical description can be found in reference [36], and the LSTM is reduced to the form of a simple operator in the following definition. The following mathematical description of the LSTM-I unit process is given:

$$\widetilde{x}_t = W_x h_{t-1} + b_x \tag{16}$$

$$x_t^c = x_t \odot m_t + (1 - m_t) \odot \widetilde{x}_t \tag{17}$$

$$h_t = LSTM(x_t^c, h_{t-1}) \tag{18}$$

$$l_t = < m_t, \pounds(x_t, \widetilde{x}_t) > \tag{19}$$

Equation (16) transforms the hidden state h_{t-1} of the previous LSTM cell into the estimated vector $\widetilde{x}_t$, where W_x and b_x are model parameters. Equation (17) replaces a missing value in the input vector x_t with the value corresponding to the estimated vector $\widetilde{x}_t$ by applying the mask vector m_t. Equation (18) generates the predicted state h_t through the LSTM network cell with x_t^c and the hidden state h_{t-1} as inputs. Equation (19) is the estimation error of the LSTM-I cell as the cumulative absolute difference between the observed and estimated values at the location of a missing value.

The encoding part of the neural network in the figure consists of a bidirectional LSTM-I neural network. An LSTM-I reads the input from the beginning to the end of the time series and generates a forward hidden-state vector sequence $\overrightarrow{h} = \{\overrightarrow{h_1}, \overrightarrow{h_2}, \ldots, \overrightarrow{h_n}\}$; the other LSTM-I reads the input backwards from the end of the time series to the beginning and produces a backward hidden-state sequence $\overleftarrow{h} = \{\overleftarrow{h_1}, \overleftarrow{h_2}, \ldots, \overleftarrow{h_n}\}$. The forward and backward hidden-state sequences are stitched together to form the encoded output $h = \{h_1, h_2, \ldots, h_n\}$ of the encoding layer, where the vector h_i is $h_i = \{\overrightarrow{h}_i, \overleftarrow{h}_i\}$.

The bidirectional LSTM-I encoding network error includes both forward and backward estimation errors.

(2) Decoder

The decoding layer receives the encoded output sequence h and produces the resulting time series sequence y. The neural network decoding process is mathematically described as Equations (20)–(22):

$$\mathrm{s_t} = \mathrm{LSTM}(h_t, \mathrm{s_{t-1}}) \tag{20}$$

$$\mathrm{y_t} = \mathrm{W_y s_t} + \mathrm{b_y} \tag{21}$$

$$l_y = < \mathrm{m_t}, \pounds(\mathrm{x_t}, \mathrm{y_t}) > \tag{22}$$

As in Equation (20), the bottom of the decoding layer is a standard LSTM network that synthesizes the encoded output sequence h to produce an output state sequence $s = \{s_1, s_2 \ldots, s_n\}$ containing valuable information. As in Equation (21), since the temperature values are continuous, a linear fully connected layer is used at the top of the decoding layer to output the imputation-based sequence y. Equation (22) gives the error of imputation results for the decoding layer, and this value is the cumulative absolute difference between the observed and interpolated values at the location of a missing value.

The error function of the entire neural network consists of three components Equation (23):

$$l_t = l_t^f + l_t^b + l_y \tag{23}$$

In Equation (23), l_t^f is the estimation error of the forward LSTM-I encoding layer, and l_t^b is the estimation error of the backward LSTM-I encoding layer.

2.4. Accuracy Evaluation

In this paper, several metrics are used to evaluate the performance of different data imputation methods, and the values of the evaluation metrics are calculated based on the test sample set. These metrics include the root mean square error (RMSE) (Equation (24)), mean absolute error (MAE) (Equation (25)), mean relative error (MRE) (Equation (26)) and Pearson correlation coefficient (PCC) (Equation (27)), which are defined as follows.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2} \tag{24}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \tag{25}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - y_i}{x_i}\right| \quad (26)$$

$$PCC = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2} \quad (27)$$

In the above index formulas, x_i is an actual missing observation in the constructed test sample set, and y_i is the interpolated result at the location of the missing value. In (27), $\overline{x}$ is the overall average of the actual observed value at the location of a missing value in the sample, and $\overline{y}$ is the overall average of the interpolated result at the location of a missing value in the formula; this value is then used to calculate the *PCC*.

3. Results and Discussion

The model implementation is based on the open source machine learning framework PyTorch (https://pytorch.org/ accessed on 20 september 2021). The training set is constructed with the observations on the left side of the missing value window for July 2020, and the test set is constructed with the observations on the right side. The deep learning imputation method includes the construction of two training samples: one with a time interval gap of 30 days and another with a time interval gap of 60 days. The acquired observations on both sides of the gap span 14 days. To distinguish these two training samples, the length of the missing value gap is used as the suffix of the corresponding deep learning imputation method below. The imputation results are shown in Figure 3, and the accuracy assessment results of various imputation methods are summarized in Table 3.

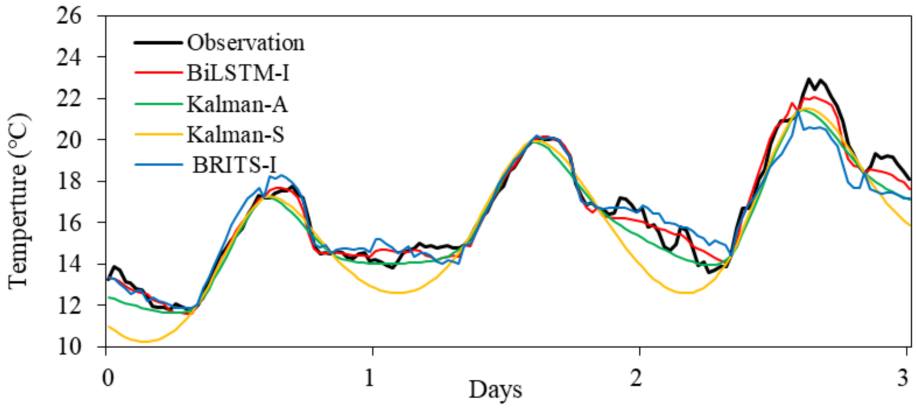

Figure 3. Comparison the imputation results of different methods with the observation data. Data for three days were randomly selected.

Table 3. Statistical table of the results of the time series imputation methods.

Method	RMSE (°C)	MAE (°C)	MRE	PCC
BiLSTM-I-60	0.4929	0.3319	0.0173	0.9963
BiLSTM-I-30	0.4686	0.3215	0.0170	0.9968
BRITS-I	1.3959	1.0300	0.0537	0.9697
Kalman-Struct	1.1742	0.8449	0.0472	0.9873
Kalman-ARIMA	0.7514	0.5469	0.0306	0.9934

Mean square error (RMSE); mean absolute error (MAE); mean relative error (MRE); pearson correlation coefficient (PCC).

A comparison of BRITS-I, the Kalman method and BiLSTM-I from Table 3 indicates that the BiLSTM-I deep learning-based imputation method developed in this paper performs best among all the methods involved. The Kalman imputation methods are better than BRITS-I. For the Kalman imputation methods, the imputation method based on the ARIMA state model yields better RMSE accuracy than Kalman-Struct. Additionally, the BRITS-I deep learning time-series imputation method is associated with the lowest accuracy (Figure 3).

3.1. BiLSTM-I vs. BRITS-I

Both BiLSTM-I and BRITS-I methods adopt the architecture of deep learning. According to Table 3, the accuracy of the BiLSTM-I model designed in this paper is higher than that of the BRITS-I model. There are two main differences between the BiLSTM-I model and the BRITS-I model. First, from the perspective of the model structure, BiLSTM-I adopts an encoder-decoder structure, and BRITS-I is equivalent to only the encoder part of the BiLSTM-I model. Such a model structure of BiLSTM-I is conducive to fully learning the potential distribution pattern of data, which can yield a high data imputation accuracy. Second, there is a difference in the model error function. The error functions of BiLSTM-I and BRITS-I consist of three parts [32]; the first two parts are the same, and the third part of the BiLSTM-I model error function involves the difference between the final estimates and true observations; therefore, the BiLSTM-I model error function evaluates the imputation results more directly, and the model convergence error and the imputation accuracy are directly related, thus ensuring that the imputation error can be minimized at the time the model converges.

3.2. BiLSTM-I vs. Kalman Smoothing

The observations on both sides of the missing value window around July 2020 are selected, and the time series decomposition equation (BSM) or ARIMA equation of state is obtained through training to establish Kalman smoothing imputation models. The accuracy of the Kalman smoothing imputation method mainly depends on whether the state equations accurately represent the time series characteristics [37]. The Kalman-S assumes that the trend and seasonal components of the time series can be fitted by the basic linear equation; the Kalman-A fits the differenced time series by establishing a regression equation. By comparison, the BiLSTM-I deep learning method does not make any assumptions about the expression form of the time series, and automatically learns the exact expression form of the time series by repeatedly training the data set to reduce the fitting errors. From the test results, the BiLSTM-I method is more likely to obtain the accurate representation of the time series than the BSM- or ARIMA-based Kalman methods, and thus obtains a higher accuracy of data interpolation.

3.3. The Generalization Ability of BiLSTM-I

The generalization ability of a model is very important in the application of deep learning methods, and the generalization ability in this paper is reflected in whether the imputation accuracy of the model is consistent for different time interval gaps. In Table 3, the missing value gaps assessed with the BiLSTM-I model are 30 days and 60 days, and the testing accuracy is basically the same for both cases, which indicates a good generalization ability. To further test this ability, we filled a time series of temperature observations with a time interval gap of 30 days by a model trained on a 60-day gap, and vice versa. The model results are shown in Figure 4, and Table 4 shows the accuracy statistics for the results of the imputation methods for these two cases. As shown in Table 4, the indicators of model accuracy are very stable for both cases, which indicates that the BiLSTM-I deep learning model has excellent generalization ability for different missing value gaps.

Table 4. Statistics for the imputation accuracy of the BiLSTM-I model applied to missing values over 30- and 60-day gaps.

Missing Value Gap	Model	RMSE	MAE	MRE	PCC
30 Days	BiLSTM-I-30	0.4686	0.3215	0.0170	0.9968
	BiLSTM-I-60	0.4865	0.3326	0.0176	0.9966
60 Days	BiLSTM-I-60	0.4929	0.3319	0.0173	0.9963
	BiLSTM-I-30	0.4834	0.3293	0.0172	0.9964

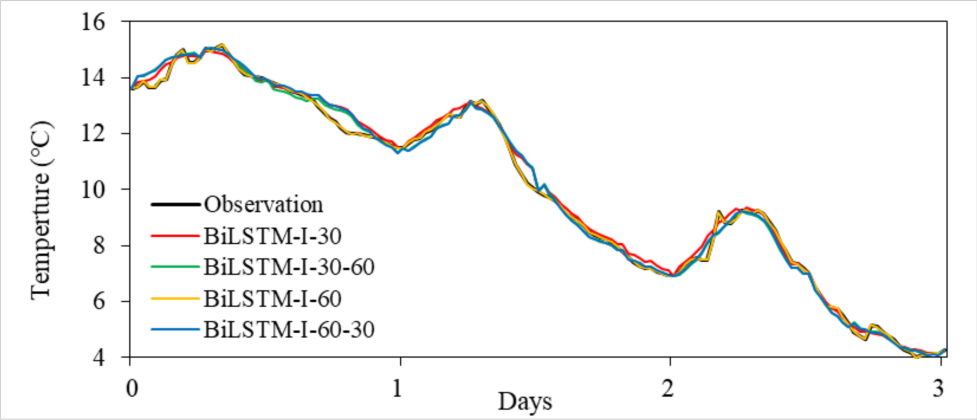

Figure 4. BiLSTM-I model results with 30- and 60-day gaps.

4. Summary

In this paper, a deep learning-based long interval gap-filling model, BiLSTM-I, was proposed for meteorological data imputation. The method addresses the practical problem of using the Seq2Seq-based deep learning technique to obtain complete high-precision, half-hourly frequency temperature observation data based on daily low-frequency temperature observations obtained manually. The experimental analysis results show that the BiLSTM-I designed in this paper outperforms other imputation methods, such as the Kalman smoothing method, or the BRITS-I deep learning method. In addition, BiLSTM-I shows great generalization ability to different missing value gaps. The RMSE for a test set with a missing value gap of 30 days is 0.47, while the RMSE for a test set with a missing value gap of 60 days is 0.49. The model not only meets the high-precision temperature data imputation requirements, but also has the potential to be applied to other meteorological dataset filling scenarios.

Author Contributions: Conceptualization, C.X. and C.H.; methodology, C.X.; software, D.Z. and W.H.; validation, C.X., C.H. and D.Z.; formal analysis, C.H.; investigation, C.X.; data curation, D.Z.; writing—original draft preparation, C.X.; writing—review and editing, C.H.; funding acquisition, C.X. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the CAS Earth Big Data Science Project, Grant No. XDA19060302; the Science and Technology Basic Resource Investigation Program of China, grant number 2017YFD0300403.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from the National Field Scientific Observation and Research Station of Dinghushan Forest Ecosystem and are available the corresponding author with the permission of the National Field Scientific Observation and Research Station of Dinghushan Forest Ecosystem.

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

References

1. Lara-Estrada, L.; Rasche, L.; Sucar, E.; Schneider, U.A. Inferring missing climate data for agricultural planning using Bayesian network. *Land* **2018**, *7*, 4. [CrossRef]
2. Huang, M.T.; Piao, S.L.; Ciais, P.; Peñuelas, J.; Wang, X.H.; Keenan, T.F.; Peng, S.S.; Berry, J.A.; Wang, K.; Mao, J.F. Air temperature optima of vegetation productivity across global biomes. *Nat. Ecol. Evol.* **2019**, *3*, 772–779. [CrossRef]
3. Hu, L.W.; He, H.L.; Shen, Y.; Ren, X.L.; Yan, S.K.; Xiang, W.H.; Ge, R.; Niu, Z.E.; Xu, Q.; Zhu, X.B. Modeling the Carbon Cycle of a Subtropical Chinese Fir Plantation Using a Multi-Source Data Fusion Approach. *Forests* **2020**, *11*, 369. [CrossRef]
4. Luedeling, E. Interpolating hourly temperatures for computing agroclimatic metrics. *Int. J. Biometeorol.* **2018**, *62*, 1799–1807. [CrossRef]
5. Afrifa-Yamoah, E.; Mueller, U.A.; Taylor, S.M.; Fisher, A.J. Missing data imputation of high-resolution temporal climate time series data. *Meteorol. Appl.* **2020**, *27*, e1873. [CrossRef]

6. Lepot, M.; Aubin, J.B.; Clemens, F. Interpolation in Time Series: An Introductive Overview of Existing Methods, Their Performance Criteria and Uncertainty Assessment. *Water* **2017**, *9*, 796. [CrossRef]
7. Carrizosa, E.; Olivares-Nadal, N.V.; Ramirez-Cobo, P. Times series interpolation via global optimization of moments fitting. *Eur. J. Oper. Res.* **2013**, *230*, 97–112. [CrossRef]
8. Schlegel, S.; Korn, N.; Scheuermann, G. On the interpolation of data with normally distributed uncertainty for visualization. *Vis. Comput. Graph.* **2012**, *18*, 2305–2314. [CrossRef]
9. Žliobaitė, I.; Hollmén, J. Optimizing regression models for data streams with missing values. *Mach. Learn.* **2015**, *99*, 47–73. [CrossRef]
10. Yang, H.M.; Pan, Z.S.; Tao, Q. Online Learning for Time Series Prediction of AR Model with Missing Data. *Neural Process. Lett.* **2019**, *50*, 2247–2263. [CrossRef]
11. Bashir, F.; Wei, H.L. Handling missing data in multivariate time series using a vector autoregressive model-imputation (VAR-IM) algorithm. *Neurocomputing* **2017**, *276*, 23–30. [CrossRef]
12. Beck, M.W.; Bokde, N.; Asencio-Cortés, G.; Kulat, K. R Package imputeTestbench to Compare Imputation Methods for Univariate Time Series. *R J.* **2018**, *10*, 218–233. [CrossRef]
13. John, C.; Emmanuel, J.E.; Nworu, C.C. Imputation of Missing Values in Economic and Financial Time Series Data Using Five Principal Component Analysis Approaches. *CBN J. Appl. Stat.* **2019**, *10*, 51–73. [CrossRef]
14. Hwang, W.S.; Li, S.Y.; Kim, S.W.; Lee, K. Data Imputation Using a Trust Network for Recommendation via Matrix Factorization. *Comput. Sci. Inf. Syst.* **2014**, *15*, 347–368. [CrossRef]
15. Tripathi, S.; Govindajaru, R.S. On selection of kernel parameters in relevance vector machines for hydrologic applications. *Stoch. Environ. Res. Risk Assess.* **2007**, *21*, 747–764. [CrossRef]
16. Sovilj, D.; Eirola, E.; Miche, Y.; Björk, K.M.; Nian, R.; Akusok, A.; Lendasse, A. Extreme learning machine for missing data using multiple imputations. *Neurocomputing* **2016**, *174*, 220–231. [CrossRef]
17. Hewamalage, H.; Bergmeir, C.; Bandara, K. Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *Int. J. Forecast.* **2021**, *37*, 388–427. [CrossRef]
18. Ma, J.; Cheng, J.C.P.; Ding, Y.X.; Lin, C.Q.; Jiang, F.F.; Wang, M.Z.; Zhai, C. Transfer learning for long-interval consecutive missing values imputation without external features in air pollution time series. *Adv. Eng. Inform.* **2020**, *44*, 101092. [CrossRef]
19. Song, W.; Gao, C.; Zhao, Y.; Zhao, Y.D. A Time Series Data Filling Method Based on LSTM-Taking the Stem Moisture as an Example. *Sensors* **2020**, *20*, 5045. [CrossRef]
20. Zhang, Y.; Zhou, B.H.; Cai, X.R.; Guo, W.Y.; Ding, X.K.; Yuan, X.J. Missing value imputation in multivariate time series with end-to-end generative adversarial networks. *Inf. Sci.* **2021**, *551*, 67–82. [CrossRef]
21. Li, Z.N.; Yu, H.; Zhang, G.H.; Wang, J. A Bayesian vector autoregression-based data analytics approach to enable irregularly-spaced mixed-frequency traffic collision data imputation with missing values. *Transp. Res. Part C Emerg. Technol.* **2019**, *108*, 302–319. [CrossRef]
22. Tsay, R.S. *Analysis of Financial Time Series*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2010.
23. Fernando, T. Kalman Filtering in R. *J. Stat. Softw.* **2011**, *39*, 1–27.
24. Einicke, G. *Smoothing, Filtering and Prediction Estimating the Past, Present and Future*; InTechOpen: London, UK, 2012.
25. Harvey, C.; Peters, S. Estimation Procedures for Structural Time Series Models. *J. Forecast.* **1990**, *9*, 89–108. [CrossRef]
26. Durbin, J.; Koopman, S.J. *Time Series Analysis by State Space Methods*; Oxford Statistical Science Series; Oxford University Press: Oxford, UK, 2001.
27. Yi, D.H. *Applied Time Series Analysis*; Renmin University of China Press: Beijing, China, 2019; pp. 18–116.
28. Xu, D.W.; Wang, Y.D.; Jia, L.M.; Qin, Y.; Dong, H.H. Real-time road traffic state prediction based on ARIMA and Kalman filter. *Front. Inf. Technol. Electron. Eng.* **2017**, *18*, 287–302. [CrossRef]
29. Jong, P.; Penzer, J. The ARIMA model in state space form. *Stat. Probab. Lett.* **2004**, *70*, 119–125. [CrossRef]
30. Tsay, R.S. State-Space Models and Kalman Filter. In *Analysis of Financial Time Series*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2005; pp. 490–542.
31. Guo, Z.J.; Wan, Y.M.; Ye, H. A data imputation method for multivariate time series based on generative adversarial network. *Neurocomputing* **2019**, *360*, 185–197. [CrossRef]
32. Cao, W.; Wang, D.; Li, J.; Zhou, H.; Li, L.; Li, Y.T. BRITS: Bidirectional Recurrent Imputation for Time Series. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018; pp. 6776–6786.
33. Lai, X.C.; Wu, X.; Zhang, L.Y.; Lu, W.; Zhong, C.Q. Imputations of missing values using a tracking-removed autoencoder trained with incomplete data. *Neurocomputing* **2019**, *366*, 54–65. [CrossRef]
34. Dabrowski, J.J.; Rahman, A. Sequence-to-Sequence Imputation of Missing Sensor Data. In *AI 2019: Advances in Artificial Intelligence*; Springer: Cham, Switzerland, 2019; pp. 265–276.
35. Zhang, Y.F.; Thorburn, P.J.; Xiang, W.; Fitch, P. SSIM-A Deep Learning Approach for Recovering Missing Time Series Sensor Data. *IEEE Internet Things J.* **2019**, *6*, 6618–6628. [CrossRef]
36. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
37. Demirhan, H.; Renwick, Z. Missing value imputation for short to mid-term horizontal solar irradiance data. *Appl. Energy* **2018**, *225*, 998–1012. [CrossRef]