International Journal of Data Science and Analytics (2024) 17:93–109
https://doi.org/10.1007/s41060-022-00358-5

REGULAR PAPER

Deep neural network-based spatiotemporal heterogeneous data reconstruction for landslide detection

Darmawan Utomo[1] · Liang-Cheng Hu[2] · Pao-Ann Hsiung[2]

Received: 17 January 2022 / Accepted: 11 August 2022 / Published online: 3 September 2022

Abstract
Landslides could cause huge threats to lives and cause property damages. In the landslide prediction system, environmental information can be collected through sensors to detect the possibility of landslide occurrences. However, the data collected by wireless sensor network systems (WSNs) may be lost due to sensor failures, external interferences, or other environmental factors, which may affect the accuracy of landslide predictions. In order to solve the problem of missing data, we propose a data reconstruction method based on rainfall intensity, soil moisture, slope, and slope direction and reconstruct missing data based on heterogeneous data and temporal and spatial relationships. A convolutional long short-term memory (ConvLSTM) deep neural network is trained to predict the missing time slot data. We use the predicted data to compensate for missing data. The results of the experiments show that the factor of safety of ConvLSTM achieves better RMSE in almost all of the missing data types and rates than LSTM. The mean and stdev forecast error of gradual fading with ConvLSTM at missing rate 30% are -0.001 and 0.033, respectively.

Keywords Landslide prediction · Data reconstruction · Neural networks · Heterogeneous data · Spatiotemporal relation · Imputation of missing data

1 Introduction

Landslide is one of the natural disasters which take lives and cause huge loss of properties [1]. Figure 1 shows the statistics of landslides that occurred in Taiwan from 2006 to 2017 [3]. Taiwan is often striked by typhoons every summer and causes serious casualties and property losses. For example, Morakot typhoon in 2009 caused serious damage to Taiwan, with 681 people killed, 33 injured and 18 missing [2]. The property loss was estimated to be more than 200 billion Taiwan dollars. If we can predict the occurrence of landslides and give early warnings, we will have more time to prepare for disaster prevention.

Much research has been devoted to landslide monitoring [4,5]. Historical data are analyzed to predict the risk of future landslides and give early warnings [6] so that people have enough time to evacuate so as to reduce the loss of property and life. Although the environmental data of landslide detection can be used to predict the occurrence of landslide, if the data are incomplete, it will affect the accuracy of prediction. Due to the instability of wireless sensor networks, data are often lost during data collection and transmission. Information related to the natural environment is constantly changing over time. In addition to communication errors and sensor errors, sensors may change their original positions due to external forces, such as during typhoon and after landslide damage. These conditions can cause changes in the feature of environmental data. Here, we propose a neural network model that uses spatiotemporal features to learn to adapt to complex and changing environmental characteristics. The goal is to address the problem of missing data and incomplete information. The accuracy of data can be improved though data reconstruction and thus reducing forecast errors.

✉ Darmawan Utomo
darmawan.utomo@uksw.edu

Liang-Cheng Hu
hlc19940420@csie.io

Pao-Ann Hsiung
pahsiung@cs.ccu.edu.tw

[1] Computer Engineering Department, Satya Wacana Christian University, Jl. Diponegoro 52-60, Salatiga 50711, Central Java, Indonesia

[2] Computer Science and Information Engineering, National Chung Cheng University, 168 University Road, Minhsiung 62102, Chiayi, Republic of China

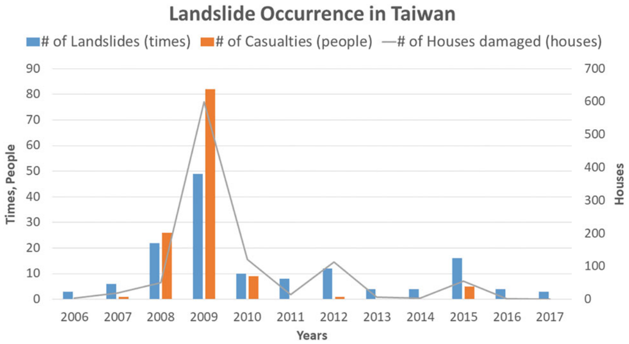

Fig. 1 Statistics of landslide occurrences in Taiwan [3]

Although some methods [9–11] have been proposed to reconstruct lost data using spatial-temporal data and heterogeneous data, there are still some problems, such as how to determine the influence of historical data and spatial relationship data on the proportion of reconstructed data separately. The nearest neighbor node is often used to reconstruct data spatially. The correlation between rainfall and soil moisture is also used to derive missing data. As environmental changes are complex, various factors interact. The above method only considers some of the factors and it is difficult to approximate the real environment. In order to reconstruct data with accuracy close to reality, a deep neural network-based data reconstruction for landslide detection method is proposed. Learning via the neural network model considers all environmental characteristics, thus improving the accuracy of data reconstruction.

Landslides occur when down-slope shear stress is larger. As shown in Equation 1 [12], the factor of safety (FS) references the stability of the soils. It takes physical properties, including rainfall, slope, and soil properties into consideration. It also requires the ability to combine physical concepts such as mechanics and hydrographic data to analyze slope stability. This equation can easily predict landslides according to the change of critical parameters and has a low computational cost, so we use FS equations model named SHALSTAB [12] to predict landslides in this work.

$$FS = \frac{C + (1 - \frac{R}{T}\frac{\alpha}{sin\theta}\frac{\rho_w}{\rho_s})\rho_s g Z cos^2\theta tan\phi}{\rho_s g Z cos\theta sin\theta} \tag{1}$$

- C: The effective coefficient (kPa)
- R: The rainfall intensity (mm/hr)
- T: The soil transmissivity (mm/hr)
- Z: The soil depth (m)
- ρ_w: The density of water (kg/m^2)
- ρ_s: The density of soil (kg/m^2)
- ϕ : The internal friction angle of the slope material ($degree$)
- θ: The slope gradient ($degree$)
- α : The specific contributing area [12].

Section 2 introduces some related work. Section 3 presents the proposed data reconstruction method. Sections 4 and 5 present the experiments and analyze the experiment results. Finally, Sect. 6 gives the conclusions and future work.

2 Related work

The need for database recovery in the database system can be divided into three main scenarios: system crash, transaction failure, and media failure. Data recovery methods to deal with the above problems include hardware, image, and signal methods. First, hardware recovery uses log-based recovery [13] that can be restarted by rescanning the log file if system crashes. Second, image recovery can be done according to the neighboring data [14–16]. Finally, signal recovery uses algorithms based on hermit and spline interpolation [17] to adjust sampling frequency.

There are several data reconstruction methods in wireless sensor networks such as interpolation methods, K-nearest neighbor (KNN) [18,19] reconstruction method that utilizes the value of the nearest K neighbor to estimate the value of the missing data. The data correlation methods [20–22] use temporal or spatial relationship to estimate missing data. Compressed sensing (CS) [10] methods are also used in data reconstruction. There are also many deep learning [23–25] methods for data reconstruction which learning the feature of dataset to reconstruct missing data. Some studies also discuss the mapping of landslide detection based on spatial and temporal factors [26,27], but after the occurrences of landslides.

The authors of a paper [8] use the deep learning with around 87,149.953 parameters with synthetic data. Tests are carried out with image by image so that the average information from the test is not visible. When tested with post-stack data at 40% random missing, it yielded RMSE of 0.094 while the reference was 0.167.

Although those methods consider the spatial or temporal relationship, they do not consider the original interactions between the data, such as the rainfall intensity will affect the soil moisture, low-lying areas are more likely to be saturated with water and so on. Reconstructing data with heterogeneous data correlation may improve the accuracy of the reconstructed data. Therefore, we will focus on reconstruction methods that have spatiotemporal and heterogeneous data correlation data.

This paper is an extended of our previous work [7]. In the previous study, the performance of the long short-term memory (LSTM) method, and linear extrapolation on the random missing data type are compared. In addition to this paper, our new work is first, we improve previous results by retraining the LSTM model from our previous work so that we get a lower average RMSE factor of safety, from 0.294 to 0.124. Second, we add the convolution long short-term memory (ConvLSTM) method and expand missing data cases in gradual fading and spatial. Finaly, we compare the prediction performance based on LSTM, and ConvLSTM for rainfall, soil moisture, method, and missing rate data types, using the retrained LSTM as the baseline.

Missing data in the WSN can occur at various layers such as the physical, network, and application layers. The causes are also various, such as environmental disturbances, sensor aging, sensor failure, battery weakening, network attacks, packet loss, human/storage errors. Improvements in the handling of missing data have an impact on several aspects such as reducing risk and cost, as well as increasing the validity of decision making. The scheme of handling the missing data is divided into methods of deletion and imputation of data [28]. In this scheme, imputation data are categorized into statistical, intelligent, and hybrid approaches. Our concern is on the intelligent approach by using the DNN time series, namely the LSTM and ConvLSTM models to replace the missing data from the prediction results.

Prediction with LSTM is applied to reconstruct missing groundwater level (GWL) data and get very good results [29]. In this approach, the data are obtained from several sensor piezometers which have a strong correlation with the missing data so that the prediction results are able to give an RMSE of 0.22m at a GWL of around 40 meters. Although the dataset used is multivariate, it comes from a homogeneous piezometer sensor.

Missing data is also possible to appear in the case of remote sensing image. The problems that arise are dead lines, and thick cloud cover. One of the imputation methods offered is to interweave convolution results from missing and original data before being trained into deep convolution neural networks (CNNs). Experimental results of [25] show that the proposed method is able to improve dead lines and thick clouds, although with limitations. This method uses 2D convolution to solve the time series problem. However, this CNN method is more pattern-based and has limited short and long-term memory capabilities as LSTM has. In addition, the case of dead lines that commonly occurs in satellite imagery is unusual in cases with landslide data. Thus, in the case of landslide, the solution for handling missing data that has WSN characteristics on landslide is interesting to study and investigate.

Therefore, in this paper we propose a solution based on time series DNN to overcome missing data on WSN with landslide character by using data from spatiotemporal heterogeneous sensors including soil moisture and rainfall. Improving the quality of reconstruction of missing soil moisture and rainfall data will improve the predictability of landslide early warning systems. Furthermore, the risk of loss of life, property, and costs can be reduced.

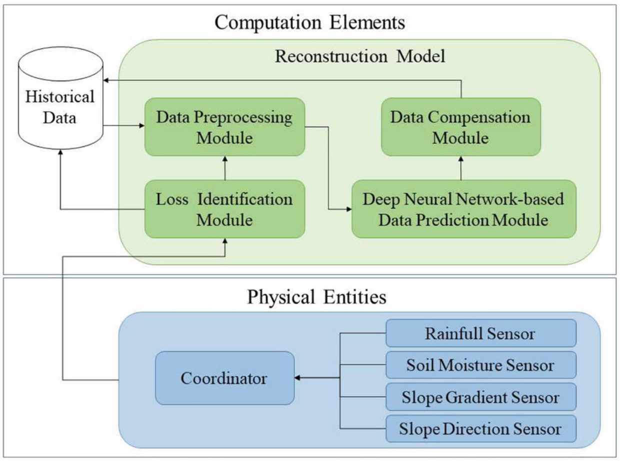

Fig. 2 Deep neural network-based data reconstruction system architecture

3 Spatiotemporal heterogeneous data reconstruction system design

A detailed introduction to our method called the spatiotemporal heterogeneous data reconstruction system (STHDRS) is described here. Figure 2 shows the basic framework of the method as well as the architectural description that we propose as follows.

The system architecture consists of two parts, including physical entities and computing elements. In the physical entity, environmental data are collected through sensor nodes. Environmental data include soil moisture, rainfall intensity, slope direction and slope. The sensed data received from sensor nodes are finally integrated by the coordinator and transmitted to the computation elements. The computation elements contains loss identification module, data preprocessing module, deep neural network-based data predication module, data compensation module and historical data and the system flowchart are shown in Fig. 3 and described in the following:

- **Loss Identification Module**
  The module detects whether the currently received environmental sensing data are missing. If the module detects that the data are missing, it records the time and location of the missing data, and if the data are normal, it is stored directly to the historical database. The specific methods will be described in Sect. 3.1.

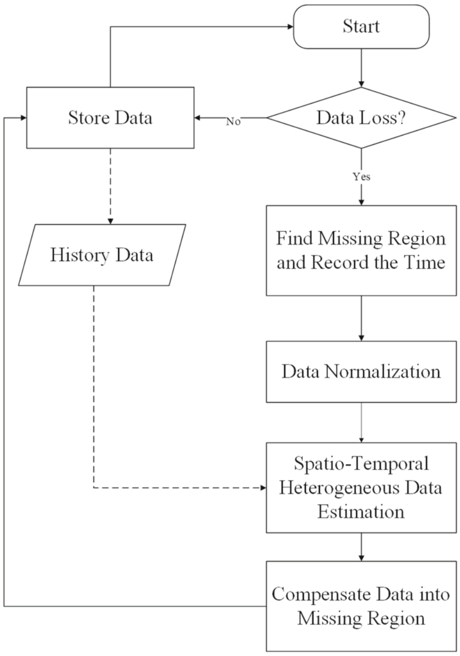
Start
Data Loss?
No
Store Data
Yes
History Data
Find Missing Region and Record the Time
Data Normalization
Spatio-Temporal Heterogeneous Data Estimation
Compensate Data into Missing Region

Fig. 3 Spatiotemporal heterogeneous data reconstruction system flowchart

- **Data Preprocessing Module**
  The task of this module is to normalize the data. It integrates four heterogeneous environmental data into a unified data format. The data and specific methods will be described in Sect. 3.2.
- **Deep Neural Network-based Data Prediction Module**
  The module reviews the temporal and spatial relationships of heterogeneous data to learn the characteristics of the environment, and predicts the future environmental data based on historical environmental data. The specific methods will be described in Sect. 3.3.
- **Data Compensation Module**
  This module is used to fill missing data. When a data loss is detected, the module will use the data predicted for the current time slot using history data to fill in the missing data. The specific methods will be described in Sect. 3.4.
- **Historical Data**
  The data reconstruction system needs to use historical data to predict the reconstruction data. Historical data contains four kinds of heterogeneous data, namely soil moisture, rainfall intensity, slope direction and slope.

The following methodologies are loss identification module, data preprocessing module, deep neural network-based data prediction module, and data compensation module which are explained as follows.

3.1 Loss identification module

This module is used to detect whether the data are complete. If the module detects a loss of data, it will record the current time and draw a location index map of the data missing node which is then used to reconstructed missing data. It provides information needed by other modules in the reconstruction model to perform reconstruction tasks. If there is no data missing, it will store the current data into history database. The loss identification algorithm is shown in Algorithm 1. First, each data type (K) in the data type set (KD) will be scanned at each node location (A) in the location set (SN), at time slot (t). If a datum is missing, the algorithm needs to record the type of data (i.e., soil moisture), the missing data location $A(i.e., (x, y))$, the time slot (t), and then set the value of the missing data location index ($M_{input}(K, A, t)$) to 1 (missing flag); otherwise, the missing data location index set to 0 (normal) and increment the number of normal nodes ($count$). When checking the missing data of a time slot is complete, it will determine whether the total number of available data ($count$) is equal to the total number of data (N) at a time slot. If they are equal, the data are complete and save to the database; otherwise, the missing data location index map is sent to the data preprocessing module.

Algorithm 1 Loss Data Identification Algorithm.

```
Input:
$D_{input}(KD, SN, t)$:The data at time $t$.
$KD$: The data type set of the sensing node.
$SN$: The location set of sensing nodes.

Output:
$D_{complete}(KD, SN, t)$: The complete data value at time $t$

Variable:
$K$: The type of data.
$A$: The location of sensor node.
$M_{input}(KD, SN, t)$: The missing data location index map at time $t$.
$N$: The total number of data in each time slot.

$count \Leftarrow 0$
for $each\ \ K\ \ in\ \ KD$ do
    for $each\ \ A\ \ in\ \ SN$ do
        if $D_{input}(K, A, t)$ is NULL then
            $M_{input}(K, A, t) = 1$
        else
            $M_{input}(K, A, t) = 0$
            $count++$
        end if
    end for
end for
if $count = N$ then
    $D_{complete}(KD, SN, t) = D_{input}(KD, SN, t)$
else
    $RunDataPreprocessing(M_{input}(KD, SN, t))$
end if
return $D_{complete}(KD, SN, t)$
```

3.2 Data preprocessing module

The scope and unit of each kind of data are different. Without a uniform benchmark, the model will not be able to learn well. Therefore, the data need to be normalized before it is input into the prediction model. The data preprocessing module is designed to handle preprocessing before data entry prediction models. The data preprocessing algorithm is shown in Algorithm 2. First, it will read the data that received from the Loss Identification Module and get the missing data time (t). Then read the historical data($D_{hisNml}(KD, SN, TS)$) for predicting the data at time point (t). To predict missing data, several data time slots are used as a prediction basis. Next, process each data type (K) in the data type set (KD), at each node location (A) in the location set (SN), at time slot (t) in the history data time slot set (TS) data, where TS is a history data time slot set that used to estimate current time slot data. Further, according to different types of data, the corresponding data preprocessing is performed. If the type of data is soil moisture (SM), historical soil moisture data in time slot set (TS) will be divided by Nml_{SM}, where Nml_{SM} is the value of maximum soil moisture minus minimum soil moisture which is called soil moisture normalization value. If the type of data is rainfall intensity (RI), historical rainfall intensity data in time slot set (TS) will be divided by Nml_{RI}, where Nml_{RI} is the value of maximum rainfall intensity minus minimum rainfall intensity which is called intensity normalization value. If the type of data is slope direction (SD), historical slope direction data in time slot set (TS) will be divided by Nml_{SD}, where Nml_{SD} is the value of maximum slope direction minus minimum slope direction which is called slope direction normalization value. If the type of data is slope rank (SR), historical slope rank data in time slot set (TS) will be divided by Nml_{SR}, where Nml_{SR} is the value of maximum slope rank minus minimum slope rank which is called slope rank normalization value. Finally, output the processed data.

Algorithm 2 Data Preprocessing Algorithm.

```
Input:
$M_{input}(KD, SN, t)$: The missing data state map at time $t$.
$D_{his}(KD, SN, TS)$: The history data of time slot set $TS$.
$KD$: The data type set of the sensing node.
$TS$: The set of history data to estimate current time slot data.
$SN$: The set of location deployed by all nodes in the sensing area.

Output:
$D_{hisNml}(KD, SN, TS)$: The normalized history data of time slot set $TS$.

Variable:
$K$: The type of data.
$A$: The location of sensor node.
$SM$: The type of soil moisture data.
$RI$: The type of rainfall intensity data.
$SD$: The type of slope direction data.
$SR$: The type of slope rank data normalization value.
$Nml_{SM}$: The soil moisture normalization value.
$Nml_{RI}$: The rainfall intensity normalization value.
$Nml_{SD}$: The slope direction normalization value.
$Nml_{SR}$: The slope rank normalization value.
Load missing data state map $M_{input}(KD, SN, t)$
Read $M_{input}(KD, SN, t)$ time
Load history data $D_{his}(KD, SN, TS)$
for each $K$ in $KD$ do
    for each $t$ in $TS$ do
        for each $A$ in $SN$ do
            if $K = SM$ then
                $D_{hisNml}(KD, SN, TS) = D_{his}(KD, SN, TS)/Nml_{SM}$
            end if
            if $K = RI$ then
    $D_{hisNml}(KD, SN, TS) = D_{his}(KD, SN, TS)/Nml_{RI}$
            end if
            if $K = SD$ then
    $D_{hisNml}(KD, SN, TS) = D_{his}(KD, SN, TS)/Nml_{SD}$
            end if
            if $K = SR$ then
    $D_{hisNml}(KD, SN, TS) = D_{his}(KD, SN, TS)/Nml_{SR}$
            end if
        end for
    end for
end for
    return $D_{hisNml}(KD, SN, TS)$
```

3.3 Deep neural network-based data prediction module

To estimate the missing data, a convolutional long short-term memory (ConvLSTM) [30] as the prediction model is employed. ConvLSTM is a powerful computation system. Although Long Short-Term Memory (LSTM) can handle time series information very well, it cannot do a good job of spatial data because spatial data has strong local characteristics. The equations for the general LSTM is as follows, where $\circ$ denotes the Hadamard product:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \tag{2}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \tag{3}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{4}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o) \tag{5}$$

$$h_t = o_t \circ tanh(c_t) \tag{6}$$

LSTM consists of input gate (i_t), forget gate (f_t), cell state (c_t), output gate (o_t), hidden state (h_t), where W is the weight matrix, b is the bias vectors, x_t is the current input data, h_t is the hidden state of memory block, and h_{t-1} is previous hidden output as shown in Equations 2 to 6. This architecture solves the problem of long-term dependencies. When new information arrives, old information is forgotten. Cell state acts as an accumulator for status information. If forget gate is active, old cell information c_{t-1} can be changed.

Unlike the general LSTM model, ConvLSTM improves the architecture of the general LSTM model by replacing the feedforward calculations with convolutions in the input-to-state and state-to-state parts of the LSTM. The general LSTM cell is shown as Fig. 4, and the ConvLSTM cell is shown as Fig. 5. The equations for the convolutional LSTM are as

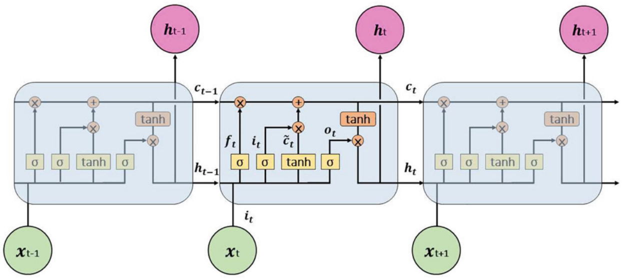

Fig. 4 The general LSTM cell

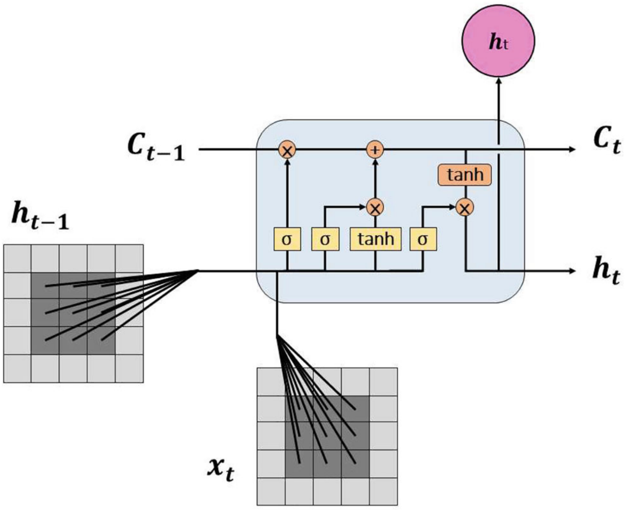

Fig. 5 The ConvLSTM cell

follows, where $*$ denotes a convolution operator:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ c_{t-1} + b_i) \quad (7)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ c_{t-1} + b_f) \quad (8)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \quad (9)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \quad (10)$$

$$H_t = o_t \circ tanh(C_t) \quad (11)$$

Therefore, ConvLSTM not only has the time-series modeling capabilities of LSTM, but also features locality like convolutional neural network (CNN). It has both temporal and spatial characteristics and also the ability to learn temporal and spatial features. Thus, this model will be used as the training model with four kinds of heterogeneous environmental data as model input. Therefore, the relationships among heterogeneous data, spatial relationships, and temporal relationships to estimate missing data are maintained. Figure 6 shows the framework of ConvLSTM-based prediction model.

The deep neural network-based data prediction module flow is shown in Fig. 7. The input data of the module are the data from the first 23 time intervals of the current time slot and are used to predict the current time slot data. The data of each time slot is composed of 4 types of data collected by 400 nodes in the sensing area, the types of data are, namely, soil moisture, rainfall intensity, slope and slope direction (the format is 20 × 20 × 4 three-dimensional data that represent the sensor are deployed in a grid of size 20 × 20 and 4 types of monitoring data). The data are trained on two layers of Con-

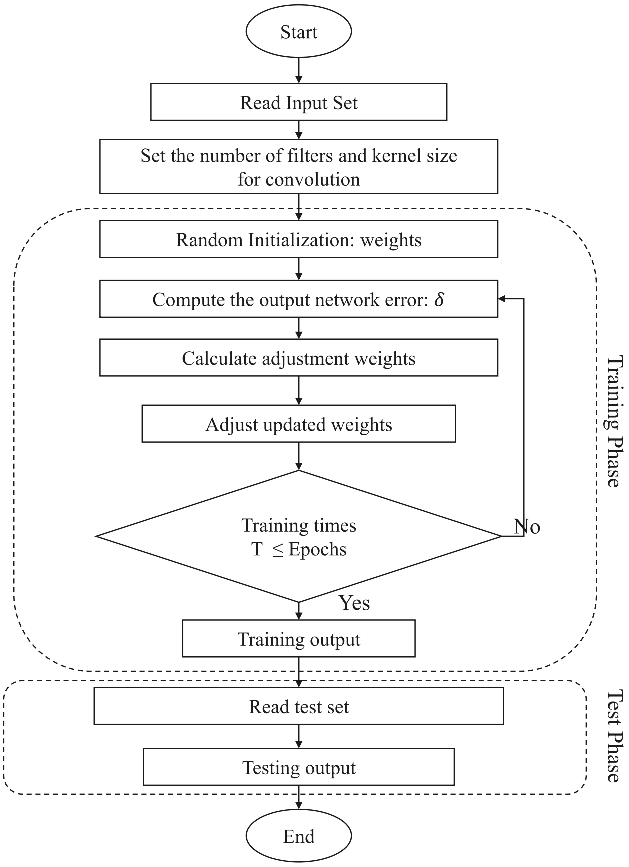
Start
Read Input Set
Set the number of filters and kernel size for convolution
Random Initialization: weights
Compute the output network error: δ
Calculate adjustment weights
Adjust updated weights
Training times T ≤ Epochs
No
Yes
Training output
Training Phase
Read test set
Testing output
Test Phase
End

Fig. 6 ConvLSTM prediction model framework design

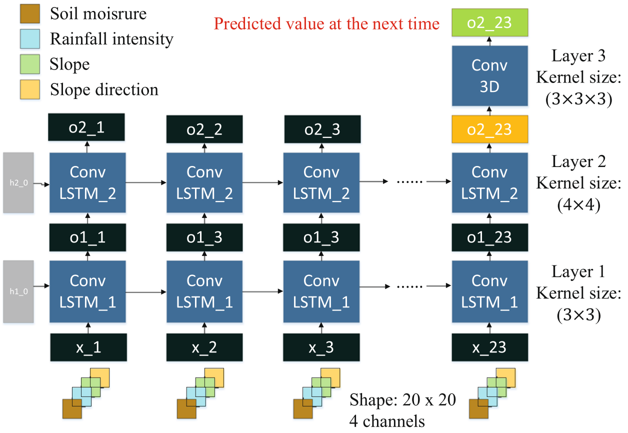

Fig. 7 The deep neural network-based data prediction module structure

Table 1 Prediction model parameters setting

Parameter	Layer 1 (ConvLSTM)	Layer 2 (ConvLSTM)	Layer 3 (Conv.)
Kernel Size	3×3	4×4	3×3 ×3
Filter number	30	40	4
Cell number	23	23	–

Table 2 Compare of different cell number

Cell number	11	17	23	29
Training time (average of each epoch)	495.5 s	505.7 s	515.8 s	526.1 s
Validation loss	9.44e-5	9.25e-5	8.55e-5	8.68e-5

Table 3 Compare different number of ConvLSTM layers

Number of ConvLSTM layers	1	2	3
Training time (average of each epoch)	230.8 s	515.8 s	889.6 s
Validation loss	1.21e-4	8.55e-5	8.53e-5

vLSTM, and then output after a convolution. This module is configured to predict a single spatial data (multivariate) with multiple data. The predicted data output format is 20 × 20 × 4 three-dimensional degree data. The parameters in our model are as shown in Table 1. The tanh activation function is used for ConvLSTM computation as expressed in Equations 12 while the sigmoid activation function which is applied for three-dimensional convolutional appeared in Equation 13.

In Table 2, some models with different numbers of cells are compared to find the optimal cells. Less training time is required when the number of cells is less. In comparison, the model using 23 cells had the lowest validation loss. In addition, the training time for all models is not much different from each other. Thus, the prediction model with 23 cells is chosen.

In Table 3, three training models with different numbers of ConvLSTM layers are compared to find the fewer the number of ConvLSTM layers that spend less time on training. A prediction model with two layers of ConvLSTM has a lower validation loss than that with one layer. The validation loss of the model with three ConvLSTM layers is slightly lower than that of the model using two ConvLSTM layers. Therefore, we believe that using two ConvLSTM layers in the model is enough to get a good prediction of missing data.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{12}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{13}$$

The difference between the predicted value and the actual value is called the prediction error when training the model. The mean square error is chosen as our loss function to calculate the error of the output during the training. It is depicted in Equation 14.

$$E = \frac{1}{N}\sum_{i=1}^{N}(T_i - O_i)^2 \tag{14}$$

where

- T_i: Target ConvLSTM data value of training sample i.
- O_i: ConvLSTM output value of training sample i.
- N: Number of training samples.

In order to reduce the prediction error, the weight of the model must be adjusted. We use Adam [31] as our optimizer to adjust weights and bias to reduce prediction errors. The Adam calculation is depicted in the following equations:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \tag{15}$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \tag{16}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{17}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{18}$$

where

- g_t: the gradients at timestep t.
- $\beta_1, \beta_2 \in [0, 1]$: the exponential decay rates.
- m_t: the exponentially decaying average of past gradients as the momentum.
- v_t: the exponentially decaying average of past squared gradients as the constraint.
- $\hat{m}_t$: the bias-corrected first moment estimate.
- $\hat{v}_t$: the bias-corrected second raw moment estimate.

The weights are defined by the following equation.

$$\theta_t = \theta_{t-1} - \frac{\alpha \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{19}$$

where

- α: learning rate.
- ϵ: the small quantity closing to zero.
- θ_{t-1}: current weights value.
- θ_t: new weights value.

The overall algorithm of deep neural network-based data prediction module is shown in Algorithm 3. First, the kernel size and number of filters are set in the ConvLSTM model. Then, set the maximum iteration number for training $Epochs$ and randomly initialize weights (W) and bias (B). The model is trained after setting parameters. Next, it will calculate the error (δ) and adjust weights (W) and bias (B). When all $Epochs$ are completed, the model training is completed. Then the trained ConvLSTM model can be used to predict the missing time slot data by input the perprocessed data $(D_{hisNml}(KD, SN, TS))$ which was output from the data preprocessing module.

Algorithm 3 Deep Neural Network-based Data Prediction Module Algorithm.

```
Input:
$D_{training}$: Environment data used for training ConvLSTM
$D_{hisNml}(KD, SN, TS)$: Inputs of data for prediction

Output:
$D_{predict}(KD, SN, t)$: Predicted output data for time $t$

Variable:
$W$: Weights of ConvLSTM
$B$: Bias of ConvLSTM
$E$: Training error between training outputs and target
$\delta$: Error value for adjusting weights and bias
$c_{sat}$: 1: Training cycle is complete, 0: Training cycle is incomplete
Set kernel size and number of filters in the ConvLSTM model
Set the maximum iteration number for training $Epochs$
Randomly initialize weights $W$ and bias $B$
//Training model
$Et \leftarrow 0$
while $(Et \leq Epochs)$ do
    $c_{sat} = 0$
    while $c_{sat} = 0$ do
        $TrainingConvLSTM(D_{training}, W, B)$
        Calculate output network error $\delta$ Equation 14
        Calculate adjustment weight $W$ and bias $B$ Equation 15 to 19
        if all samples are trained then
            $c_{sat} = 1$
        end if
    end while
    $Et++$
end while
//Do Prediction
$D_{predict}(KD, SN, t) = RunConvLSTM(D_{hisNml}(KD, SN, TS))$
return $D_{predict}(KD, SN, t)$
```

3.4 Data compensation module

This module is designed to fill in the missing data area after obtaining an estimated data from the deep neural network-based data prediction module for the time that data were missing. Next is to find out the corresponding missing area data estimated data based on the missing data state map. Then compensate the estimate data into missing area after data denormalization. The data compensation algorithm is shown in Algorithm 4. In the flowchart, first, load missing data $(D_{input}(KD, SN, t))$, missing data index map $(M_{input}(KD, SN, t))$ and predicted data $(D_{predict}(KD, SN, t))$. Then pick out the missing area predict data according to missing data location index map $(M_{input}(KD, SN, t))$ in each data type (K) in the data type set (KD), at each node location (A) in the location set (SN), at time slot (t). Further, inverse the normalized according to the corresponding data type. If the type of data is soil moisture (SM), predicted soil moisture data in time slot t will be multiplied by Nml_{SM}. If the type of data is rainfall intensity (RI), predicted rainfall intensity data in time slot t will be multiplied by Nml_{RI}. If the type of data is slope direction (SD), predicted slope direction data in time slot t will be multiplied by Nml_{SD}. If the type of data is slope rank (SR), predicted slope rank data in time slot t will be multiplied by Nml_{SR}. Finally, fill the predicted data into missing data and output the reconstructed data $(D_{reconstruct}(KD, SN, t))$.

4 Experiments

The proposed spatiotemporal heterogeneous data reconstruction system for data reconstruction evaluation is presented here. First, the experimental datasets used for experiments are introduced. Then, the experiment results are illustrated.

4.1 Experimental datasets

The soil moisture and rainfall intensity data from the historical environment monitoring datasets used in this research are from the Soil & Water Conservation Bureau, Council of Agriculture, Taiwan [3] . The time interval of the rainfall intensity and soil moisture data is five minutes. The slope direction and slope rank data are from the National Land Surveying and Mapping Center (NLSC), Taiwan [33]. Sensor nodes are deployed in a grid of size 20 × 20 and the distance between each sensors is around 2 km. The data used in this work were collected from April 5, 2013 to October 12, 2017. First, 20,000 data samples were used for training the LSTM model, then 5,000 samples were used for validating the model. Finally, 300 samples were applied as testing stage.

Algorithm 4 Data Compensation Algorithm.

```
Input:
$D_{input}(KD, SN, t)$: The data at time $t$
$M_{input}(KD, SN, t)$: The missing data location index map at time $t$.
$D_{predict}(KD, SN, t)$: Predicted output data at time $t$
$KD$: The data type set of the sensing nodes.
$SN$: The location set of sensing nodes.

Output:
$D_{reconstruct}(KD, SN, t)$: The reconstruct data at time $t$

Variable:
$K$: The type of data.
$A$: The location of sensor nodes.
$SM$: The type of soil moisture data.
$RI$: The type of rainfall intensity data.
$SD$: The type of slope direction data.
$SR$: The type of slope rank data.
$Nml_{SM}$: The soil moisture normalization value.
$Nml_{RI}$: The rainfall intensity normalization value.
$Nml_{SD}$: The slope direction normalization value.
$Nml_{SR}$: The slope rank normalization value.
Load time $t$ data $D_{input}(KD, SN, t)$
Load missing data state map $M_{input}(KD, SN, t)$
Load time $t$ prediction data $D_{predict}(KD, SN, t)$
for each $K$ in $KD$ do
    for each $A$ in $SN$ do
        $D_{predict}(KD, SN, t) = M_{input}(KD, SN, t) \times D_{predict}(KD, SN, t)$
        if $K = SM$ then
            $D_{predict}(KD, SN, t) = D_{predict}(KD, SN, t) \times Nml_{SM}$
        end if
        if $K = RI$ then
            $D_{predict}(KD, SN, t) = D_{predict}(KD, SN, t) \times Nml_{RI}$
        end if
        if $K = SD$ then
            $D_{predict}(KD, SN, t) = D_{predict}(KD, SN, t) \times Nml_{SD}$
        end if
        if $K = SR$ then
            $D_{predict}(KD, SN, t) = D_{predict}(KD, SN, t) \times Nml_{SR}$
        end if
        $D_{reconstruct}(KD, SN, t) = D_{predict}(KD, SN, t) + D_{input}(KD, SN, t)$
    end for
end for
return $D_{reconstruct}(KD, SN, t)$
```

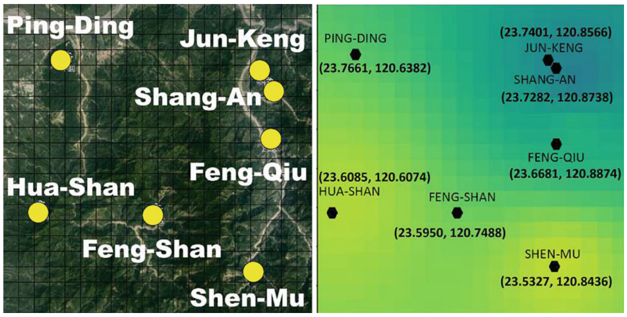

Fig. 8 Satellite (left) and soil moisture maps with station locations (right)

All of the data were collected from the Shen-Mu, Shang-An, Jun-Keng, Feng-Qiu, Hua-Shan, Feng-Shan and Ping-Ding stations. The location of satellite imagery from seven stations is shown in Fig. 8, on the left. To illustrate an area, the seven stations are interpolated. The right Fig. 8 displays the soil moisture image as a result of the development using the inverse distance weighting equation [34].

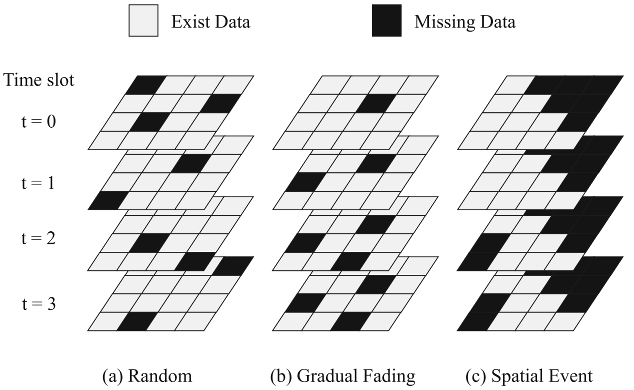

Fig. 9 Types of missing data

Not all data are provided by CWB over a four-year period, but only data due to heavy rain and typhoon events. This is because landslides are generally only sensitive to rainfall or earthquakes. The total data provided include 11 heavy 22 typhoons and rains, named Trami, Kong-rey, Usagi, Fitow, Hagibis, Aere, Nesat, Matmo, Fung-wong, Chan-hom, Soudelor, Goni, Dujuan, Koppu, Nepartak, Meranti, Soulik, Cimaron, Malakas, Megi, Haitang, Hato, and Guchol. The missing raw data which are marked as -9999 are reconstructed manually using a linear interpolation method.

4.2 Type of missing data

As shown in Fig. 9, three types of missing data are considered in this work, including random missing, gradual fading, and spatial event. These missing data types are used to evaluate the efficiency of the proposed reconstruction method and to compare it to existing data reconstruction methods.

- Random
  Random missing simulates the situation where data are missing due to signal interference, transmission error, and data missing.

- Gradual fading
  Any sensor's lifetime is limited and the sensor will gradually fade until the energy is exhausted. In addition, the sensor nodes are deployed in an external environment and can often be damaged by external environmental factors. A broken sensor node will result in the inability to send the sensed data to the coordinator.

- Spatial event
  Spatial events simulates the loss of sensors in an entire area due to the occurrence of earthquakes and landslides. Lost sensor nodes will no longer transmit data.

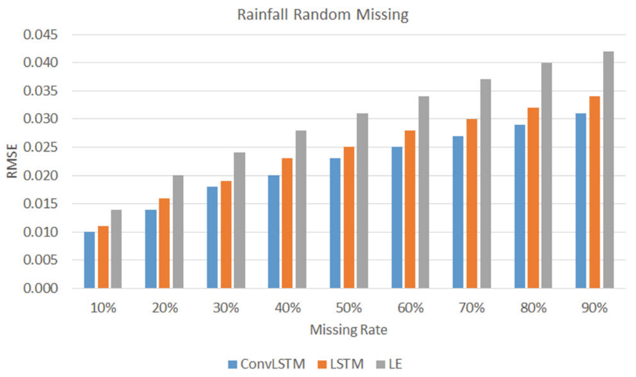

Fig. 10 Comparing different reconstruction methods for random missing rainfall data

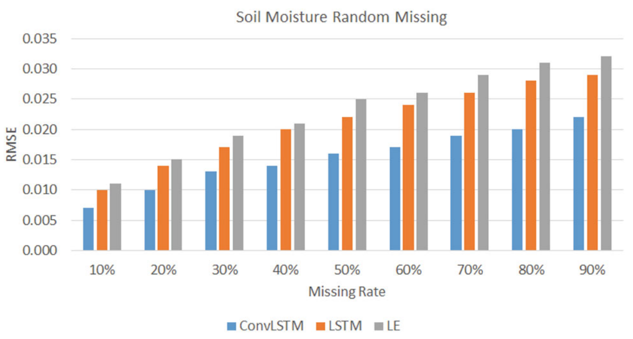

Fig. 11 Comparing different reconstruction methods for random missing soil moisture data

5 Experiment results

The reconstruction accuracy is evaluated by the metric root mean squared error (RMSE) as shown in Equation 20. It is frequently used to evaluate the difference between two sets of values. The smaller the value, the closer are the two values. This formula is used as an assessment for the similarity between the reconstructed data values and the original data values. Since the slope and the slope direction data exhibit almost no change with time, in this experiment, the rainfall and moisture data are used to evaluate the reconstructed data.

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}(y-\widehat{y})^2} \tag{20}$$

- n: the numbers of time slots for which data was missing.
- y: the original soil moisture data value.
- $\widehat{y}$: the reconstructed soil moisture and rainfall data values.

The ConvLSTM and LSTM methods are compared with the linear extrapolation (LE) method as the baseline method. Two previous points are applied to predict the next point. This point should give a good result because the prediction point is close to the previous point. Equation 21 shows how to calculate the LE.

$$L_n = 2L_{n-1} - L_{n-2} \tag{21}$$

- L_n: Linear extrapolation of the missing point
- $_n$: n-th time stamp

5.1 Evaluation for the random missing data type

In this section of experiment, the reconstruction of data in the random missing data type including (1) comparing different reconstruction methods for random missing rainfall data and (2) comparing different reconstruction methods for random missing soil moisture data will be evaluated.

In the first experiment, the rainfall data for random missing data type are reconstructed and then evaluated using the RMSE for each missing rate. As shown in Fig. 10, the performance of LE method is the worst, while the other two deep learning methods can learn the trend of historical data so the reconstruction results are better. In addition, Fig. 10 also shows comparing with different reconstruction methods for random missing rainfall data. In here, our method performance obtains the best.

In the second experiment, the soil moisture data in random missing data type are reconstructed and then calculated the RMSE for each missing rate. As shown in Fig. 11, the performance of LE method is the worst, while the other two deep learning methods can learn the trend of historical data so the reconstruction results are better. Compare with Fig. 10, Since the rainfall is 0 for most of the time, the calculated rainfall RMSE is lower than the RMSE of the soil moisture. As shown in Fig. 11, comparing with different reconstruction methods for random missing soil moisture data, our method performance is the best. As shown in Fig. 12, the data reconstructed by the proposed method is closer to the original data than the others methods. The RMSE value of ConvLSTM in the time slot 148 is 0.0077 which is the lowest than LE and LSTM that give 0.0362 and 0.0313, respectively.

The following mean and standard deviation of original data are two statistical data for soil moisture and rainfall data on LSTM and ConvLSTM. The (mean, stdev) of rainfall data and soil moisture is (0.7638, 0.098) and (0.1315, 0.1047), respectively. These values are from normalized data. The average value of soil moisture is 0.7638, while in rainfall it is 0.1315. The character of the data from soil moisture tends to change slowly with the relative humidity value above 0.60. On the other hand, the character of the rainfall data is very dynamic with relatively wide rainfall. Heavy rain will produce high and temporary spike data. When it does not rain, the rainfall becomes 0. So it is difficult for the rainfall predictor to produce a small RMSE. In the experiments shown in Figs. 10 and 11, the RMSE of soil moisture is lower than that of rainfall.

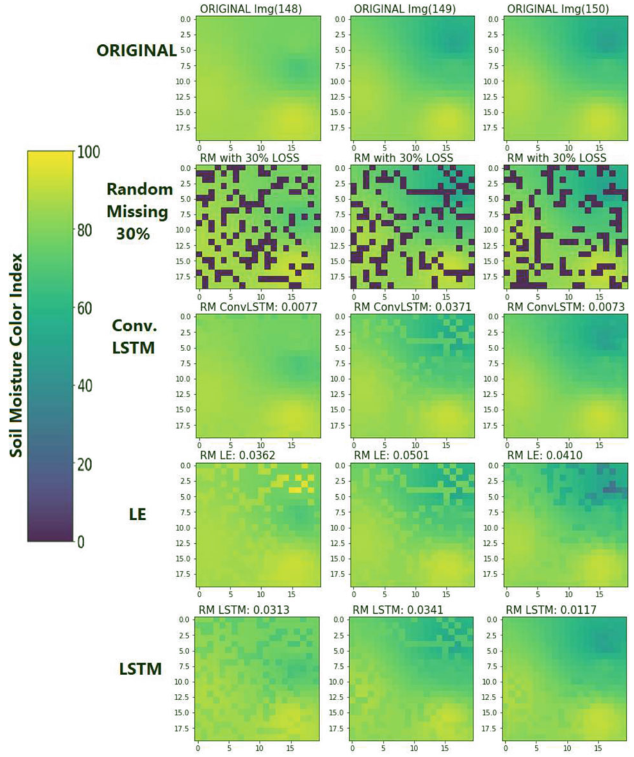

Fig. 12 Comparing different reconstruction methods for 30% missing rate random missing soil moisture data

In the experiment [8] uses 87,149,953 parameters with synthetic datasets called pre-stack and post-stack. When tested with post-stack data at 40% random missing, it yielded RMSE of 0.094 while the reference was 0.167.Our experimental LSTM and ConvLST parameters are 50,800 and 317,142, respectively. Much simpler than [8]. The average RMSE from the results of the rainfall data test with a missing rate of 40% with testing data that was not trained for LSTM and ConvLST were 0.023 and 0.021, respectively. The results of this comparison can provide an illustration that our proposed design is faster and simpler and provides better quality.

5.2 Evaluation for the gradual fading data type

In this section of experiment, the reconstruction of data in the gradual fading data type including (1) comparing different reconstruction methods for gradual fading rainfall data and (2) comparing different reconstruction methods for gradual fading soil moisture data are evaluated. The rainfall and soil moisture average fading time slots of each missing rate are shown in Table 4.

In the first experiment, the rainfall data in gradual fading data type are reconstructed and the RMSE is calculated for each missing rate. As shown in Fig. 13, the performance of LE is the worst. When a node is fading or even dies over time, we can observe that gradual fading has strong temporal characteristics. Therefore, the deep learning model with learning time series features has achieved good results. For all the missing rates of 10% to 90%, the RMSE is the smallest when ConvLSTM is used. As shown in Fig. 13, comparing with different reconstruction methods for gradual fading rainfall data, ConvLSTM method performance is the best.

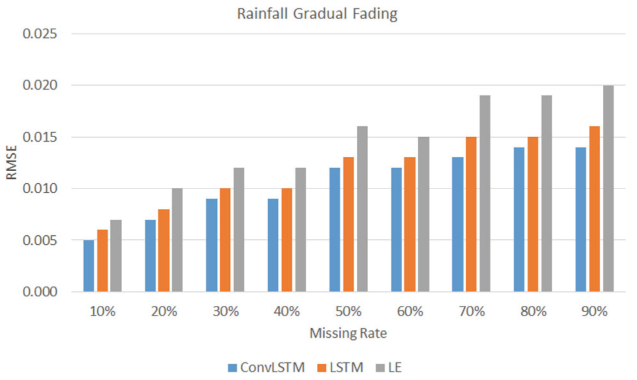

Fig. 13 Comparing different reconstruction methods for gradual fading rainfall data

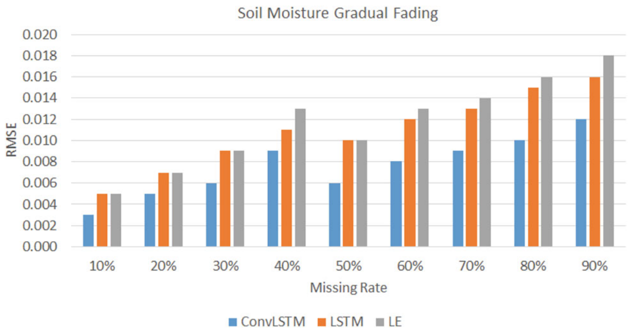

Fig. 14 Comparing different reconstruction methods for gradual fading soil moisture data

In the second experiment, the soil moisture data in Gradual Fading data type are reconstructed, and the RMSE is calculated for each missing rate. As shown in Fig. 14, the performance of LE is the worst. Both of LSTM and ConvLSTM reconstruction result are good because the characteristics of learning temporal correlation in deep learning model. The performance of ConvLSTM method is the best. Compare with Fig. 13, since the rainfall is 0 for most of the time, the calculated rainfall RMSE is lower than the RMSE of the soil moisture. As shown in Fig. 15, the data reconstructed by ConvLSTM method in average is closer to the original data due to the features that ConvLSTM can learning not only temporal but also spatial correlation of data. Although the RMSE of LSTM is the best in the time slot 149, in other time slots, ConvLSTM shows better.

Table 4 Missing rate in gradual fading data type

Missing rate (%)	Total number of broken sensor nodes	Average fading (time slots) rainfall	Average fading (time slots) soil moisture
10	40	187	164
20	80	201	200
30	120	190	213
40	160	210	197
50	200	187	200
60	240	194	196
70	280	204	192
80	320	199	211
90	360	194	193

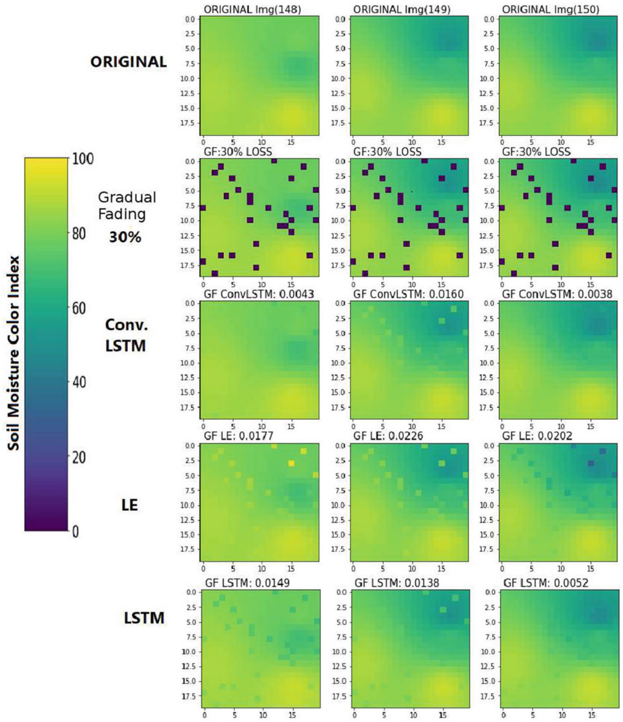

Fig. 15 Comparing different reconstruction methods for gradual fading soil moisture data

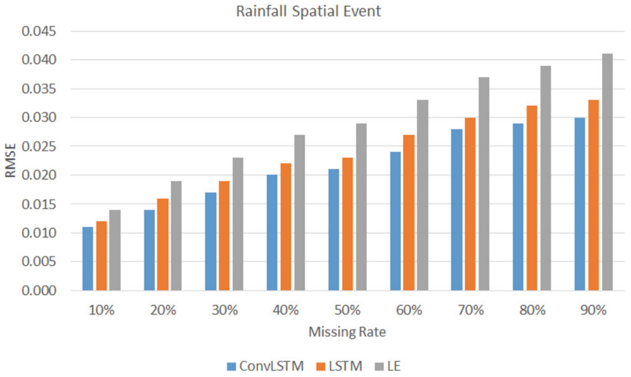

Fig. 16 Comparing different reconstruction methods for spatial event rainfall data

5.3 Evaluation for the spatial event data type

In this section of experiment, the reconstruction of data in the spatial event data type including (1) comparing different reconstruction methods for spatial event rainfall data and (2) comparing different reconstruction methods for spatial event soil moisture data is evaluated.

In the first experiment, the rainfall data in spatial event data type are reconstructed and the RMSE is calculated for each missing rate. As shown in Fig. 16, it can be seen that the effect of reconstruction by ConvLSTM method is the best because its RMSE is smallest. In addition, comparing with different reconstruction methods for spatial event rainfall data, this method performance is the best.

In the second experiment, the soil moisture data in spatial event data type are constructed, and the RMSE is calculated for each missing rate. As shown in Fig. 17, it can be seen that the effect of reconstruction by our proposed method is the best because its RMSE is smallest. As shown in Fig. 17, comparing with different reconstruction methods for spatial event soil moisture data, the proposed method performance is the best. As shown in Fig. 18, we can see that our method can reconstructed data well even in the case of large areas without data for a long time. The average RMSE of time slots 148 up to 150 for ConvLSTM and LSTM is 0.0109 and 0.0208, respectively. This short examples shows ConvLSTM performs better than LSTM.

5.4 Evaluate the impact of reconstruction data on landslide prediction

In this section, the reconstructed soil moisture and rainfall data are used to estimate the occurrence of landslides and verified the impact of reconstruction data on landslide predictions by comparing the difference between the original

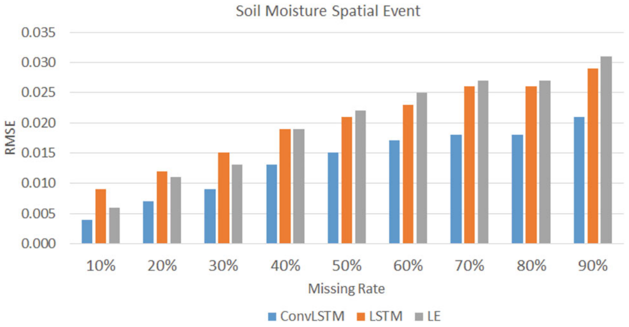

Fig. 17 Comparing different reconstruction methods for spatial event soil moisture data

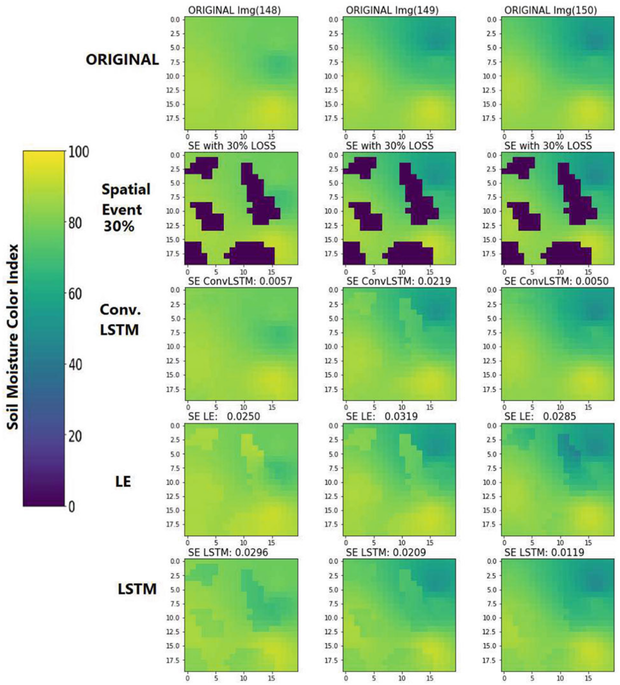

Fig. 18 Comparing different reconstruction methods for spatial event soil moisture data

data and the reconstructed data FS using RMSE. The factor of safety (FS) [12,35] equation is applied here as prediction model for landslide as shown in Equation 22. Some of the following parameters are assumed according to the paper: The cohesion (C) and internal friction angle (phi) are related to soil moisture from [36]. The bulk density $(\rho_s) = 2.5\ g/cm^3$ and transmissivity $(T) = 4.28\ m^2/day$ from [12] and assume that water content is 1.9 kg when soil moisture is reached to 100%. The soil depth (Z) is changed with the slope which is referenced from [37]. The Equation 23 is applied to soil cohesion and Equation 24 for internal friction angle.

$$FS = \frac{C + (1 - \frac{R}{T}\frac{\alpha}{sin\theta}\frac{\rho_w}{\rho_s})\rho_s g Z cos^2\theta tan\phi}{\rho_s g Z cos\theta sin\theta} \tag{22}$$

$$C = 12752\theta^2 - 3722.9\theta + 293.15 \tag{23}$$

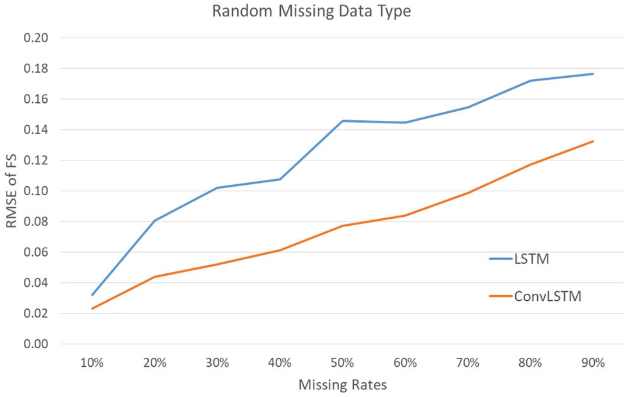

Fig. 19 Comparing the RMSE of FS for different reconstruction methods for random missing data type

$$\phi = -3661.5\theta^2 + 610.64\theta + 15.707 \tag{24}$$

- C: soil cohesion (kPa)
- ϕ: internal friction angle $(degree)$
- θ: water content $(kg \cdot kg^{-1})$

In this section of experiment, the RMSEs of calculated FS are evaluated from the reconstructed data in three missing data types including (1) comparing the RMSE of FS for different reconstruction methods for random missing data, (2) comparing the RMSE of FS for different reconstruction methods for gradual fading data, (3) comparing the RMSE of FS for different Reconstruction methods for spatial event data, and (4) comparing forecast error on random missing, gradual fading, and spacial event data types at missing rate 30%. Missing rate is defined as the increase in the amount of data loss over a given time. A missing rate of 30% means that a total of 120 data were lost over a 24-hour period. The time of data loss is randomly distributed.

In the first experiment, the data in random missing data type are reconstructed, and then the FS is calculated, and further, the RMSE is calculated for each missing rate. As shown in Fig. 19, it can be seen that the RMSE of ConvLSTM is always shown better than LSTM in all of the experiments. On average, the ConvLSTM is 38.5% better than LSTM.

In the second experiment, the data in gradual fading data type are reconstructed, and then the FS is calculated, and further, the RMSE is calculated for each missing rate. As shown in Fig. 20, it can be seen that only on the missing rate 10% of the LSTM RMSE is a little bit better than ConvLSTM. However, on average, the ConvLSTM is 17.1% better than LSTM.

In the third experiment, the data in spatial event data type are reconstructed, and then the FS is calculated, and further, the RMSE is calculated for each missing rate. As shown in Fig. 21, it can be seen that the RMSE ConvLSTM is always

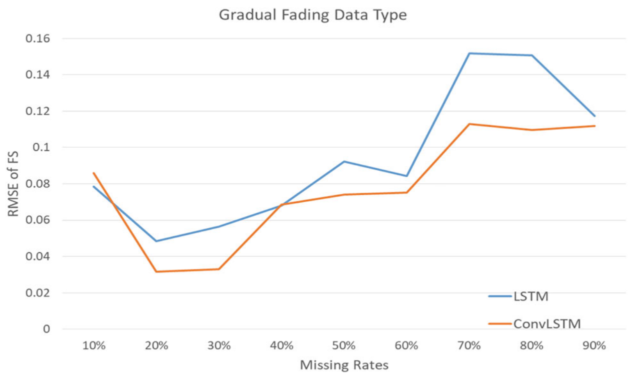

Fig. 20 Comparing the RMSE of FS for different reconstruction methods for gradual fading data type

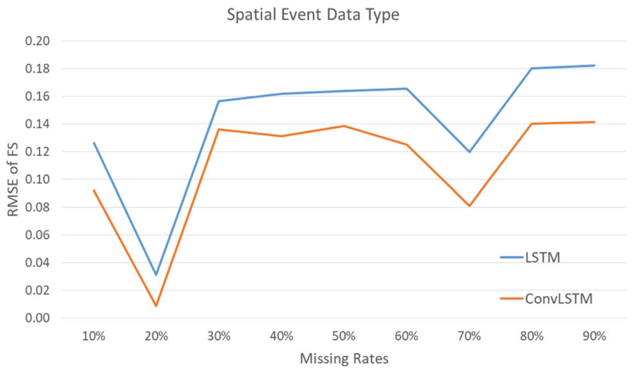

Fig. 21 Comparing the RMSE of FS for different reconstruction methods for spatial event data type

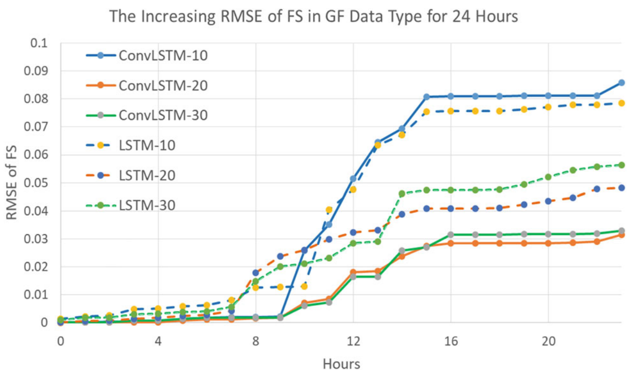

Fig. 22 Comparing the increasing RMSE of gradual fading FS for different reconstruction methods and missing rates

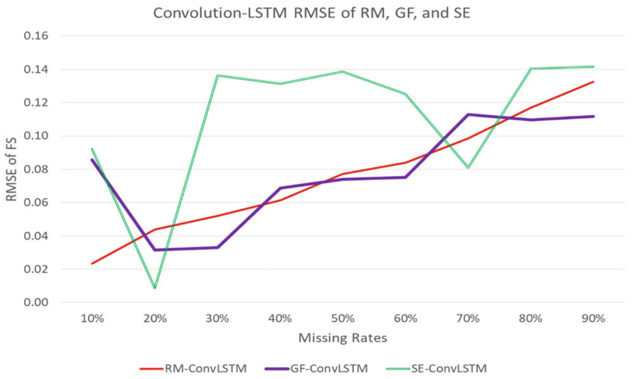

Fig. 23 Comparing the RMSE of convolution LSTM's FS for different data types

better than LSTM. On average, ConvLSTM is 27.6% better than LSTM despite the unsmooth graphs.

In general, an increase in the missing rate will result in an increase in RMSE as shown in the random missing case. However, in the case of gradual fading and spatial event, it appears that there are unsmooth graphs. Unsmooth graphs can occur due to latent errors that propagate to the next prediction. There are two main possibilities as a result of the prediction results as a result of these latent errors. The first possibility is a predictor that gives results that differ greatly from the reference. This results in the resulting error will always be carried over, especially in the gradual fading and spatial event data types. In the random missing data type, the location of the missing data is not fixed so that latent error propagation does not occur. The second possibility is a predictor that gives a result that is close to the reference value. This small latent error is propagated to the next prediction. As a result, the RMSE of the second possibility is a smaller total RMSE than the first.

In Fig. 22, the propagation of latent error in the gradual fading data type with missing rates from 10 to 30% is shown hourly. The solid lines and dashed lines represent the ConvLSTM and LSTM methods, respectively. The number of errors is seen to increase from the $0th$ hour to the $23rd$ hour which represents latent errors and new additional errors. The latent error at the missing rate of 10% is the result of the first possibility, causing a drastic increase from the $9th$ hour to the $14th$ hour. The two methods, ConvLSTM and LSTM, indicate that they both generate large errors and propagate latent errors. In this case, ConvLSTM gives a higher error than LSTM while at 20% and 30% missing rates, ConvLSTM propagates lower error than LSTM.

Justification for the existence of large or small error propagation in gradual fading and spatial events is proposed by comparison with the random missing data type. The random missing data type shows a graph that consistently increases in proportion to the missing rate. Figure 23 shows the ConvLSTM graph of the three data types. From this graph, it can be seen that a large error propagation appears in most of the spatial events. Meanwhile error propagation from gradual fading swings around random missing. From Fig. 23, it can be seen that there are a total of 11 and 7 points of gradual fading and spatial events which have errors above and below the random missing data type, respectively.

The fourth experiment is an effort to find and compare forecast errors on random missing, gradual fading, and spa-

Table 5 Mean and standard deviation of different methods and data type at missing rate = 30%

Type method	Mean	Stdev
RM-LSTM	−0.003	0.102
RM-ConvLSTM	−0.004	0.052
GF-LSTM	−0.001	0.056
GF-ConvLSTM	−0.001	0.033
SE-LSTM	0.000	0.157
SE-ConvLSTM	−0.010	0.136

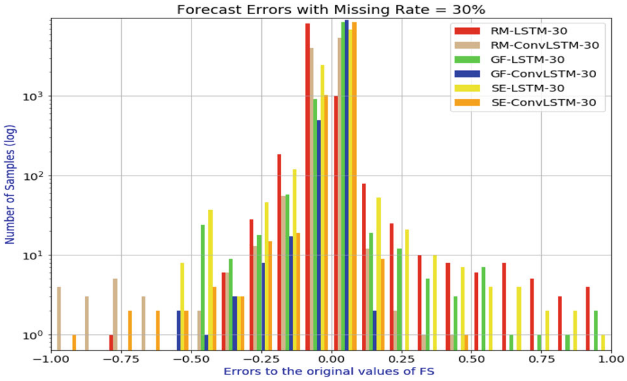

Fig. 24 RMSE forecast error for different reconstruction methods and data types at missing rate 30%

cial event data types at missing rate 30%. The histogram graph in Fig. 24 shows the forecast error of the three data types. There are 21 bins that are worth -1 to +1 with 0.1 steps. On the y-axis, it is scaled logarithmically. In general, the forecast error generated from this graph is dominant in the bin from -0.2 to 0.2. The mean and Stdev of this forecast error are shown in Table 5. In general, the average of this forecast error is around 0.0 while the Stdev is from 0.033 to 0.157 which are GF-ConvLSTM and SE-LSTM, respectively. From Fig. 24, the shortest and longest bin ranges are shown in the GF-ConvLSTM (blue) and SE-LSTM (orange) bar graphs, respectively.

The results of these four experiments show that ConvLSTM is very promising with better performance than LSTM in almost all of data types. In Fig. 23 can be used as an indicator of the parts of the data that need to be retrained and retained. Data that produces the first possibility or error that is greater than the random missing data type needs to be retrained. On the other hand, for prediction results that produce a second possibility or a smaller error than the random missing data type, these data must be retained. In this way, the performance of this system will get better following the development of better data.

Table 6 Comparisons of time consumption and resource usage

	Training Time (s)	GPU usage	Memory usage
LSTM	3970 s	36%	5807 MB
ConvLSTM	14023 s	88%	5907 MB

Table 7 Comparisons of reconstruction time consumption

CPU	i7-4720	i7-11700
GPU	–	RTX-3080
LSTM	3.6 ms	0.9 ms
ConvLSTM	10.8 ms	1.9 ms

5.5 Evaluate the consumption of data reconstruction system

The time consumption and resource usage of CovnLSTM and LSTM are compared here. These data are collected when running the training model and the results are expressed in Table 6. It is shown that the ConvLSTM training process requires more GPU resources, memory and takes more time to train than LSTM does. Larger resource requirements are a weakness of the ConvLSTM method compared to LSTM. Therefore, in this training process it is highly recommended to use the GPU.

Furthermore, the results of this training process are weights and biases that will be used during the inference process. The inference process generally does not require heavy computing so it can be delegated to machines with CPU only or CPU+GPU if faster results are desired. Table 7 shows the comparison of the reconstruction times of the two methods. Here, the reconstruction processes using the ConvLSTM method takes more time than that carried out by the LSTM. The average time required by ConvLSTM on the CPU is 10.8 ms, but the sensor data sampling rate is five minutes per data. Thus, the reconstruction time of the ConvLSTM method can still satisfy real-time data reconstruction and can be applied to an early warning system for landslides.

6 Conclusion

The spatial-temporal heterogeneous data reconstruction system for landslide detection has been proposed here. The factor of safety is applied to verify the impact of the reconstructed data on landslide prediction. The heterogeneous data (such as rainfall intensity, soil moisture, slope, and slope direction) are processed to reconstruct missing rainfall and soil moisture data. Three types of missing data, namely random missing, gradual fading, and spatial event, are considered as the possible causes. The prediction performance between the historical environmental data and the reconstructed data is evaluated by using RMSE. The experimental results show that ConvLSTM is very promising with better RMSE performance than LSTM in almost all missing data types. In addition, although ConvLSTM requires longer training and inference than LSTM, the average reconstruction time required is about 2 ms which is much less than the five minutes average sensor data sampling.

In addition, in the future, more forecasting methods such as a combination between statistics and DNN that show promising results and the impact of sensor anomalies on our proposed approach will be considered. If abnormal data are used as model training data, it will probably reduce the accuracy of data reconstruction. Therefore, determining the anomaly data and classify it as an error, and reconstruct it would increase the performance. In addition, in order to increase the accuracy of the reconstruction data, switching the different data reconstruction models according to the seasonality of the climate will be observed. .

Acknowledgements This paper is an extended version of our paper entitled Reconstruction of deep neural network-based data for landslide detection published in the 2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). This work is supported by the Ministry of Science and Technology, Taiwan, R.O.C., for financial support of this research under project numbers MOST 140-2221-E-194-064

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Dai, F., Lee, C., Ngai, Y.: Landslide risk assessment and management: An overview. Eng. Geol. **64**(1), 66–87 (2002)
2. Typhoon morakot, https://en.wikipedia.org/wiki/Typhoon_Morakot, 2022
3. Soil and water conservation bureau,'https://246.swcb.gov.tw, 2022
4. Musaev, A., Wang, D., Pu, C.: Litmus: a multi-service composition system for landslide detection. IEEE Trans. Serv. Comput. **8**(5), 715–726 (2015)
5. Wang, B.: A landslide monitoring technique based on dual-receiver and phase difference measurements. IEEE Geosci. Remote Sens. Lett. **10**(5), 1209–1213 (2013)
6. Ramesh, M.V., Rangan, V.P.: Data reduction and energy sustenance in multisensor networks for landslide monitoring. IEEE Sens. J. **14**(5), 1555–1563 (2014)
7. Utomo, D., Hu, L.-C., Hsiung, P.-A.: Deep neural network-based data reconstruction for landslide detection, in *IGARSS 2020 - 2020 IEEE international geoscience and remote sensing symposium*, pp. 3119–3122 (2020)
8. Chai, X., Gu, H., Li, F., Duan, H., Hu, X., Lin, K.: Deep learning for irregularly and regularly missing data reconstruction. Sci. Rep. **10**(1), 3302 (2020)
9. Xiang, L., Luo, J., Rosenberg, C.: Compressed data aggregation: energy-efficient and high-fidelity data collection. IEEE/ACM Trans. Netw. **21**(6), 1722–1735 (2013)
10. Kong, L., Xia, M., Liu, X.Y., Wu, M.Y., Liu, X.: Data loss and reconstruction in sensor networks, in *Proceedings of the IEEE conference on computer communications* pp. 1654–1662 (2013)
11. Wang, C., Cheng, P., Chen, Z., Liu, N., Gui, L.: Practical spatiotemporal compressive network coding for energy-efficient distributed data storage in wireless sensor networks, in *Proceedings of the IEEE vehicular technology conference* pp. 1–6 (2015)
12. Huang, J.C., Kao, S.J., Hsu, M.L., Liu, Y.A.: Influence of specific contributing area algorithms on slope failure prediction in landslide modeling. Nat. Hazard. **7**(6), 781–792 (2007)
13. Strom, R.E., Yemini, S.: Optimistic recovery in distributed systems. ACM Trans. Comput. Syst. **3**(3), 204–226 (1985)
14. Chen, B., Huang, B., Chen, L., Xu, B.: Spatially and temporally weighted regression: a novel method to produce continuous cloud-free landsat imagery. IEEE Trans. Geosci. Remote Sens. **55**(1), 27–37 (2017)
15. Zhang, K., Gao, X., Tao, D., Li, X.: Multi-scale dictionary for single image super-resolution, in *Proceedings of the IEEE conference on computer vision and pattern recognition* pp. 1114–1121 (2012)
16. Qin, Y., Wang, F.: A curvature constraint exemplar-based image inpainting, in *Proceedings of the international conference on image analysis and signal processing* pp. 263–267 (2010)
17. Li, J., Cheng, S., Gao, Z.: Approximate physical world reconstruction algorithms in sensor networks. IEEE Trans. Parallel Distrib. Syst. **25**(12), 3099–3110 (2014)
18. Cover, T., H, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
19. Zhang, M.L., Zhou, Z.H.: A k-nearest neighbor based algorithm for multi-label classification, in *Proceedings of the IEEE international conference on granular computing*, Vol. 2, pp. 718–721 (2005)
20. Nower, N., Tan, Y., Lim, A.O.: Efficient spatial data recovery scheme for cyber-physical system, in *Proceedings of the IEEE international conference on cyber-physical systems, networks, and applications* pp. 72–77 (2013)
21. Nower, N., Tan, Y., Lim, A.O.: Efficient temporal and spatial data recovery scheme for stochastic and incomplete feedback data of cyber-physical systems, in *Proceedings of the IEEE international symposium on service oriented system engineering*pp. 192–197 (2014)
22. Nower, N., Tan, Y., Lim, Y.: Incomplete feedback data recovery scheme with kalman filter for real-time cyber-physical systems," in *Proceedings of the 7th international conference on ubiquitous and future networks* pp. 845–850 (2015)
23. Shi, W., Jiang, S., Zhao, D.: Deep networks for compressed image sensing, in *Proceedings of the IEEE international conference on multimedia and expo (ICME)* pp. 877–882 (2017)
24. Mousavi, A., Baraniuk, G.B.: Learning to invert: signal recovery via deep convolutional networks, in *Proceedings of the IEEE international conference on acoustics, speech and signal processing (ICASSP)* pp. 2272–2276 (2017)
25. Zhang, Q., Yuan, Q., Zeng, C., Li, X.: Wei, Y.: Missing data reconstruction in remote sensing image with a unified spatial-temporal-spectral deep convolutional neural network, *IEEE transactions on geoscience and remote sensing* pp. 1–15 (2018)
26. He, S., Tang, H., Li, J., Tang, J., Li, S.: Landslide detection with two satellite images of different spatial resolutions in a probabilistic topic model, in *2015 IEEE international geoscience and remote sensing symposium (IGARSS)* pp. 409–412 (2015)
27. Qingqing, H., Yu, M., Jingbo, M., Anzhi, Y., Lei, L.: Landslide change detection based on spatio-temporal context, in *2017 IEEE international geoscience and remote sensing symposium (IGARSS)* pp. 1095–1098 (2017)

28. Adhikari, D., Jiang, W., Zhan, W., He, Z., Rawat, D.B., Aickelin, U., Khorshidi, H.A.: A comprehensive survey on imputation of missing data in internet of things. ACM Comput. Surv. (2022). https://doi.org/10.1145/3533381
29. Vu, M., Jardani, A., Massei, N., Fournier, M.: Reconstruction of missing groundwater level data by using long short-term memory (lstm) deep neural network. J. Hydrol. **597**, 125776 (2021)
30. Shi, X.J., Chen, Z.R., Wang, H., Yeung, D.Y., Wong, W.K., Wang, C.W.: Convolutional lstm network: a machine learning approach for precipitation nowcasting, in *In proceedings of the conference on neural information processing systems (NIPS)* (2015)
31. Kingma, D.P., Ba, D.P.: Adam: A method for stochastic optimization, *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980
32. Soil and water conservation bureau platform for monitoring data, http://monitor.swcb.gov.tw, 2018
33. National land surveying and mapping center for monitoring data, https://maps.nlsc.gov.tw/, 2018
34. Geography, G.: Inverse distance weighting idw interpolation, https://gisgeography.com/inverse-distance-weighting-idw-interpolation/, 2022
35. Huang, J.C., Kao, S.J.: Optimal estimator for assessing landslide model performance. Hydrol. Earth Syst. Sci. **10**(6), 957–965 (2006)
36. Machado, A.L.T., Trein, C.R.: Characterization of soil parameters of two soils of Rio Grande do Sul in modeling the prediction of tractive effort. Eng. Agrícola. **33**(4), 709–717 (2013)
37. Soil depth in Taiwan, http://sdl.ae.ntu.edu.tw/TaiCATS/knowledge_detail.php?id=54, 2022

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.