A Deep-Learning-Based Forecasting Ensemble to Predict Missing Data for Remote Sensing Analysis

Monidipa Das, *Student Member, IEEE*, and Soumya K. Ghosh, *Member, IEEE*

***Abstract*—The problem of *missing data* in remote sensing analysis is manifold. The situation becomes more serious during *multitemporal analysis* when data at various a-periodic timestamps are missing. In this work, we have proposed a deep-learning-based framework (*Deep-STEP_FE*) for reconstructing the missing data to facilitate analysis with remote sensing time series. The idea is to utilize the available data from both earlier and subsequent timestamps, while maintaining the *causality constraint* in spatiotemporal analysis. The framework is based on an *ensemble of multiple forecasting modules*, built upon the observed data in the time-series sequence. The coupling between the forecasting modules is accomplished with the help of dummy data, initially predicted using the earlier part of the sequence. Then, the dummy data are progressively improved in an iterative manner so that it can best conform to the next part of the sequence. Each of the forecasting modules in the ensemble is based on Deep-STEP, a variant of the deep stacking network learning approach. The work has been validated using a case study on predicting the missing images in *normalized difference vegetation index* time series, derived from *Landsat-7 TM-5 satellite imagery* over two spatial zones in *India*. Comparative performance analysis demonstrates the effectiveness of the proposed forecasting ensemble.**

***Index Terms*—Causality constraint, deep learning, ensemble, missing data, prediction, remote sensing.**

I. Introduction

REMOTE sensing data play a significant role in land cover change analysis, climate change detection, anthropogenic impacts analysis, ecosystem monitoring, and so on [1], [2]. However, one of the common obstacles, often appearing in remote sensing time-series analyses, is the nonavailability of data in the temporal sequence. The origin of such missing information is the unavailable source satellite imagery, which mainly happens due to low temporal frequency, defective sensor, poor atmospheric condition, or other image-specific problems. A more grave situation arises in a multitemporal analysis when all the complementary spatial information for a particular time instant is missing. Such an example has been depicted in Fig. 1. It shows a sequence of normalized difference vegetation index (NDVI) images, derived from Landsat-7 TM-5 raw satellite imagery, where the image at time instant $(t+3)$ is missing in the sequence from time t to $(t+5)$. This may hinder the subsequent interpretation, leading to inefficient performance of any analytical process, like urban sprawl detection, land cover change prediction, and so on. It is, therefore, necessary to somehow retrieve such missing images in order to facilitate the further analyses with these data.

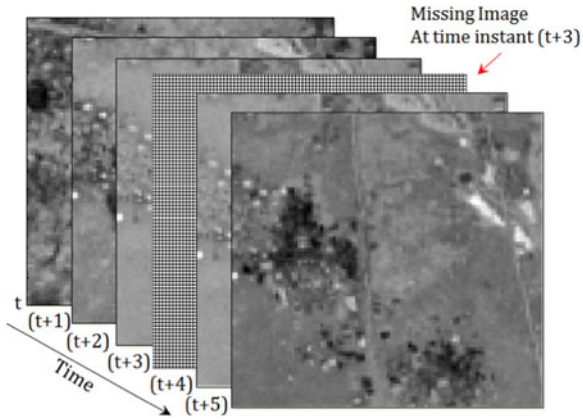

Fig. 1. Missing image in the sequence of NDVI imagery.

In this work, we aim at predicting such *derived* (single band) missing imagery by utilizing the available spatial information at the other time instants in the given sequence, without disobeying the *causality constraint* in spatiotemporal (ST) analysis. As per the causality constraint, the conditioning neighborhood should consist of only the data from earlier timestamps [3]. According to Snepvangers *et al.* [4], using values from future events to explain the past is "conceptually awkward" and can lead to "physically unrealistic" results, especially when sudden inputs in the system appear. Similar situation arises while using backcasting models (prediction in backward direction) or using forward–backward ensembles [5] during prediction. In order to maintain the causality constraint, one may choose to use only past measurements for missing value prediction; however, this fails to properly utilize all the available information, especially those from future time instants.

In our work, we attempt to utilize the available values even from the future timestamps, without contradicting with the causality constraint. Our proposed Deep-STEP_FE is an ensemble of a number of forecasting modules (based on Deep-STEP [6]), each of which uses the consecutively available time-series data (image) from past time instants to predict the immediate next missing data (image) in the sequence. Each of these predicted data items is further progressively tuned in an iterative

Manuscript received March 14, 2017; revised July 27, 2017; accepted September 19, 2017. Date of current version November 27, 2017. *(Corresponding author: Monidipa Das.)*

The authors are with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721302, India (e-mail: monidipadas@hotmail.com; skg@iitkgp.ac.in).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2017.2760202

manner so that it best conforms to the next part of the entire sequence. During the data tuning process also, we apply forecasting (prediction in forward direction) to maintain the causality constraint. While tuning, all the predicted data at intermediate time instants are used along with the other available data in the sequence to predict the data for the latest available timestamp. The predicted data for this final timestamp are compared with the original data of the corresponding timestamp. The process continues until the prediction accuracy reaches a particular level of satisfaction. Since the overall prediction process is performed in the forward direction and the feature set of the training samples is prepared using the data from the past time instants, the causality constraints are maintained. The available spatial information from future is only used to check the conformity of the predicted image with the corresponding original one.

A. Related Work and Motivation

Several techniques for predicting or reconstructing missing data in the remote sensing imagery have been proposed till date. Depending on the source of complementary information used for missing data reconstruction, these techniques can be categorized into four groups: *spatial methods*, *spectral methods*, *temporal methods*, and *hybrid methods* [7].

The *spatial methods* work without any other auxiliary information source. The propagated diffusion methods [8], spatial interpolation methods [9], etc., are some widely used spatial methods for regenerating the missing information in remote sensing analysis. However, since, in spatial methods, the reconstruction is done using the remaining data in the same image only, these methods cannot utilize the other data available in the temporal sequence. Therefore, in the present context of multitemporal analyses, the spatial methods are comparatively less useful. The *spectral methods* [10] utilize the spectral domain as the source of complementary information. These are suitable for predicting missing data in multispectral and hyperspectral imagery, having redundant spectral information. However, these methods also fail to utilize the information available for the other time instants. On the other hand, the *temporal methods* use the data acquired at different time periods (at the same position) as the complementary information. The temporal replacement methods, temporal learning models [11], etc., are some popular methods of this kind. However, even with the available spatial information, the temporal methods cannot utilize these for missing data prediction. Therefore, various *hybrid methods* have recently been proposed, which attempt to make better use of the information hidden in the spatial, spectral, and temporal dimensions. The combined ST [9] and spatiospectral [12] methods belong to this category.

Among these hybrid techniques, the spatiospectral methods are not suitable for multitemporal analysis. On the other side, the ST methods, like STAR, STARMA [3], ST kriging [9], STMRF [7], etc., are commonly used for ST prediction purpose. However, computation with a very large dataset containing several thousands of records/features becomes a challenging issue for most of these techniques. Moreover, the models such as STAR, STARMA, etc., follow the *causality constraint*, which restricts these techniques to use data from the subsequent part of the temporal sequence.

In this work, we have proposed a *deep-learning*-based *hybrid ST model* to predict the missing image for multitemporal remote sensing analysis. In our work, we are dealing with the data *derived* from satellite remote sensing imagery containing millions of pixels. In order to extract the intrinsic spatial/ST features/patterns from such voluminous data, we need some advanced data mining techniques. Since deep learning offers efficient algorithms that can learn in multiple levels corresponding to different levels of abstractions, we have chosen the same in our present work of ST prediction. Our proposed model is an ensemble of multiple forecasting modules, each based on the Deep-STEP approach [6]. The objective behind the use of Deep-STEP as a constituent forecasting module is that the Deep-STEP has been able to show encouraging performance in ST prediction of remote sensing data containing several thousands to millions of pixels [6]. However, one of the limitations of the Deep-STEP is that, similar to the STAR and STARMA techniques, it is built on the base of *causality constraint*, which confines Deep-STEP to take advantage of the ST information in the subsequent part of the time series. Motivated by this fact, the present work proposes an ensemble model (*Deep-STEP_FE*), where the coupling between the Deep-STEP-based forecasting modules is done in such a way that all the available data, both in the earlier and subsequent timestamps with respect to the missing image, are properly utilized without contradicting with the constraint.

B. Problem Statement and Contributions

The overall problem addressed in the present paper can be formally defined as follows.

1) Given a sequence of derived remote sensing imagery $I_1, \ldots, I_{m-1}, ?, I_{m+1}, \ldots, I_{x-1}, ?, I_{x+1}, \ldots, I_t$ over any variable for t number of timestamps, where the sign "?" indicates the missing images of some a-periodic timestamps, the problem is to predict or reconstruct the missing images $I_m, \ldots, I_x, \ldots$, considering the same ST framework.

In this regard, our major contributions are as follows.

1) Proposing a *forecasting ensemble* model (*Deep-STEP_FE*) to predict missing data for remote sensing analysis, by utilizing available ST data from *both* earlier and subsequent timestamps.
2) Exploring *deep-learning-based* ST analysis techniques for missing image reconstruction purpose.
3) Overcoming the limitation of the Deep-STEP approach [6] while maintaining the *causality constraints* in spatial time-series analysis.
4) Validating the proposed *Deep-STEP_FE* approach in predicting missing NDVI imagery (derived from Landsat-7 TM-5 raw satellite time series [13] over two spatial zones in *India*), each containing several thousands of pixels.

The rest of this paper is organized as follows. The architecture and learning procedure of our proposed deep-learning-based forecasting ensemble has been illustrated in Section II. The experimentation and results are discussed in Section III. Finally, the concluding remarks have been made in Section IV.

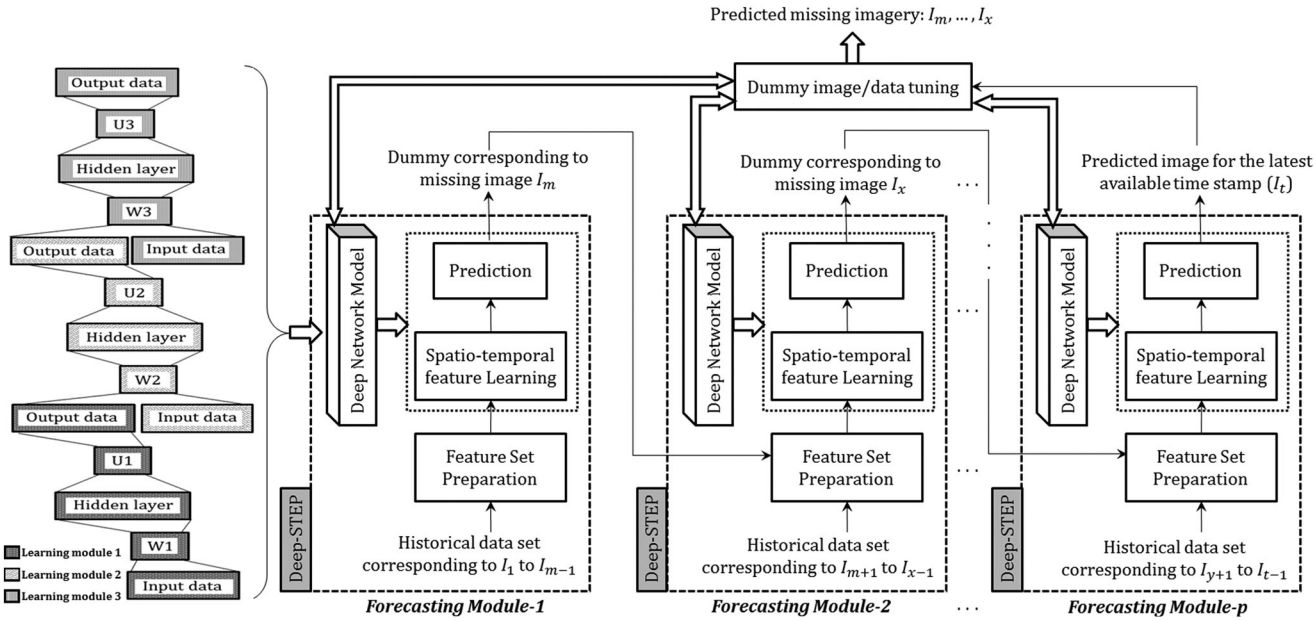

Fig. 2. Proposed forecasting ensemble model: Deep-STEP_FE.

II. Deep-STEP_FE: The Proposed Deep-Learning-Based Forecasting Ensemble Model

This section provides the details of our proposed forecasting ensemble, termed as *Deep-STEP_FE*. The building block of *Deep-STEP_FE* is the Deep-STEP approach [6], which is influenced from the deep stacking network (DSN) architecture. Two of the significant properties of DSN learning are: 1) even for large datasets, the DSN does not require GPUs; and 2) the DSN is able to utilize supervision information at every module. The Deep-STEP approach inherits the above-mentioned properties of the DSN, which, in turn, facilitates the presently proposed *Deep-STEP_FE* model as well.

The overall architecture of the proposed ensemble model is depicted in Fig. 2. The architecture is defined in such way that it conforms to the problem statement, described in Section I-B. As shown in the figure, the proposed *Deep-STEP_FE* is an ensemble of p number of forecasting modules, where $(p-1)$ is the total number of missing images in the time series of available remote sensing imagery. Each forecasting module is based on the Deep-STEP approach, which works with the consecutively available datasets from the earlier timestamps and predicts the immediately next missing image as a dummy image so as to provide a continuous set of imagery for the entire duration. With the help of dummy data corresponding to the missing imagery, the final forecasting module predicts the image of the latest available timestamp and checks for the accuracy. The process continues in an iterative manner until there is no improvement in the final prediction, while tuning the dummy images in each iteration.

The entire architecture is composed of *four* fundamental blocks: *Feature set preparation, ST feature learning, prediction*, and *dummy image/data tuning*. Each of these functional blocks is illustrated in the subsequent part of the section.

A. Feature Set Preparation

The objective here is to represent each pixel in the available imagery in terms of ST features. The feature set preparation is based on the assumption that the intensity of each pixel $I(x,y,t)$ in the image raster is a function of intensity of its ST neighbors [3]. Numerically, it can be expressed as follows:

$$I(x,y,t) = \psi(I(x+\Delta x_i, y+\Delta y_j, t+\Delta t_k)) \qquad (1)$$

where $(\Delta x_i, \Delta y_j, \Delta t_k)$ expresses the ST neighborhood coverage with $\Delta t_k < 0$, which follows the *causality constraint* in time-series analysis.

Since the Deep-STEP-based forecasting modules in the proposed forecasting ensemble take into account the temporal evolution of each pixel, the feature set for a pixel $I(x,y,t)$ at time t is prepared only with the neighboring pixels at time $t-1$, including itself. Therefore, considering maximum *spatial coverage of neighbor* to be s (i.e., $\Delta x_i = \Delta y_j = s$), the feature set for a pixel at (x,y) location becomes

$$\{I(x-s, y-s, t-1), \ldots, I(x,y,t-1), \ldots \\ \ldots, I(x+s, y+s, t-1)\}.$$

In case the image at time $(t-1)$ is not available (refer to forecasting modules 2 to p in Fig. 2), the corresponding dummy image, as generated by the previous forecasting module, is used to prepare the feature set.

In this manner, the input dataset of dimension $[P \times Q]$ is prepared for each time instant t separately, where $Q = (2s+1)^2$ is the size of input feature vector corresponding to each of the P observed pixels at time t.

B. Spatiotemporal (ST) Feature Learning

The ST feature learning within each forecasting module in the proposed ensemble is based on the deep network model used in the Deep-STEP approach.

1) Deep Network Architecture: The deep network architecture is an integral part of Deep-STEP, which has been used in each of the forecasting modules within the proposed ensemble model (Deep-STEP_FE). The central idea of this architecture is the concept of stacking, like that in the DSN [14], where the simple modules of functions are composed first, and then, these are stacked on top of each other for learning complex functions. A simplified architecture of the network model with only three such modules is shown in the left-hand side of Fig. 2. Each of the feature learning modules in the deep network model is basically perceptron with single layers of hidden neurons and has been denoted by distinct patterns/shades. However, in an actual prediction scenario, the total number of learning module in the network will be equal to the total number of consecutive training images available for the corresponding forecasting module. Thus, if the number of training images is increased, the number of modules and, hence, the depth of the architecture will also increase.

The output of each learning module (except the top most one) is the learned spatial feature set for the corresponding time instant, whereas the input to each learning module is the combined raw spatial feature set as prepared in the above-discussed manner (see Section II-A) and the spatial feature set learned by the previous learning module. In case of the first forecasting module, the input to the bottom-most learning module is only the raw spatial feature set corresponding to the first available image in the sequence. The output of the top-most module is the predicted value at a given pixel location in the corresponding missing image. After the required processing, these values are used as the dummy values to aid the feature set preparation in the subsequent forecasting module. In case of the final forecasting module, the output of the top-most learning module is the predicted value at a given pixel location in the latest available image (I_t) in the sequence.

2) Deep Network Learning: Let, for any learning module in the deep network model, the training vectors are denoted by $X = [x_1, x_2, \ldots, x_P]^T$, in which each input vector x_i is of dimension D and is denoted by $x_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$, and P is the total number of training samples. Also, let $H = [h_1, h_2, \ldots, h_P]^T$ denote the activity matrix over all training samples in the hidden layer, let L denote the number of hidden units, and let C denote the output vector dimension for any learning module. Then, the output of any learning module can be expressed as follows

$$Y = \sigma\big(\sigma\big(XW^T\big)U^T\big) \tag{2}$$

where U is a $[C \times L]$-dimensional weight matrix at the upper layer within the module, W is an $[L \times D]$-dimensional weight matrix at the lower layer within the same module, and $\sigma(.)$ is a sigmoid function. Moreover, as per the Deep-STEP approach, the value of D for the bottom-most module is Q, whereas that for all the higher level modules is $2Q$.

Under this deep learning architectural setting, the weight matrices W and U in each learning module are learned as per the Deep-STEP approach in the following manner.

Let the target vectors over P samples be $T = [t_1, t_2, \ldots, t_P]^T$, where each $t_i = [t_{i1}, t_{i2}, \ldots, t_{iC}]$. Then, the cost function is as follows:

$$J = -\frac{1}{P}\sum_{i=1}^{P}\sum_{j=1}^{C}[t_{ij}\log y_{ij} + (1 - t_{ij})\log(1 - y_{ij})] + G \tag{3}$$

where G is the regularization term computed as follows:

$$G = \frac{\lambda}{2P}\left[\sum_{i=1}^{L}\sum_{j=1}^{D} w_{ij}^2 + \sum_{i=1}^{C}\sum_{j=1}^{L} u_{ij}^2\right] \tag{4}$$

where λ is the regularization parameter; u_{ij} and w_{ij} are the elements in the ith row and the jth column in the matrices U and W, respectively.

Then, as per Deep-STEP, the gradient calculation for the cost function is performed using backpropagation in the following way.

For the lower layer weights, we have

$$\frac{\partial J}{\partial W} = \frac{1}{P}\left[\left[((Y - T)U) \circ \sigma'\big(XW^T\big)\right]^T X\right] + \frac{\lambda}{P}W \tag{5}$$

$$= \frac{1}{P}\left[\left[((Y - T)U) \circ \sigma'\big(XW^T\big)\right]^T X + \lambda W\right] \tag{6}$$

where

$$\sigma'\big(XW^T\big) = \sigma\big(XW^T\big) \circ \big(1 - \sigma\big(XW^T\big)\big) \tag{7}$$

and symbol "$\circ$" denotes elementwise multiplication.

Similarly, for the upper layer weights in the module, we have

$$\frac{\partial J}{\partial U} = \frac{1}{P}\left[(Y - T)^T\sigma\big(XW^T\big)\right] + \frac{\lambda}{P}U \tag{8}$$

$$= \frac{1}{P}\left[(Y - T)^T\sigma\big(XW^T\big) + \lambda U\right]. \tag{9}$$

During the learning process, the output Y of each preceding learning module is merged with the input feature set corresponding to the immediate next time instant and then the merged output is fed as the set of input vectors in the succeeding learning module (see Fig. 2). It helps to incorporate the temporal evolution of the pixels/data points in the feature learning process. Moreover, the input dataset X is normalized so that it remains consistent with the learning module outputs, which are forced to be in the range $[0, 1]$, because of using the binary sigmoid function ($\sigma(.)$).

C. Prediction

On the basis of the updated weight values and ST features learned at the bottom levels, the top-most layer in the deep network within each forecasting module generates a P-dimensional vector Y, denoting the predicted values. Since the predicted values are obtained in normalized form within the range $[0, 1]$, these are further mapped to the original scale by means of *linear stretching* to obtain the actual prediction values. The predicted values from the final or pth forecasting module are used to

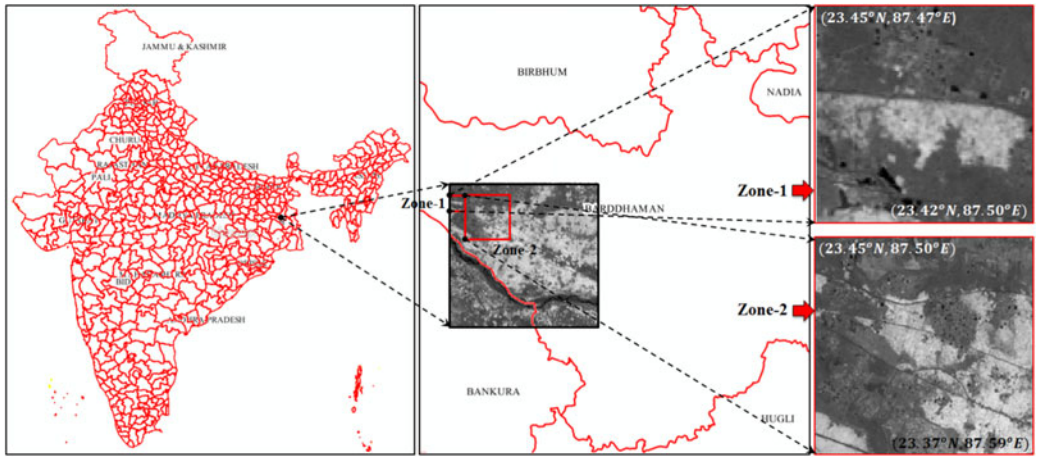

Fig. 3. Study area in the district of Barddhaman, West Bengal, India.

check the stopping condition of the forecasting process in the proposed *Deep-STEP_FE*, whereas the predicted values from the other forecasting modules are reshaped into the original image size and are used as dummy values to form feature sets in the subsequent forecasting modules. This helps in utilizing the available data of all the timestamps without contradicting the causality constraint in ST analysis.

D. Dummy Image/Data Tuning

The objective of this functional block is to control the overall forecasting process within the proposed forecasting ensemble, ensuring progressive improvement of each of the dummy images (corresponding to missing imagery) in an iterative manner. Once the predicted image is obtained from the final (pth) forecasting module, it is compared with the original image (I_t) of that particular timestamp. If the accuracy of the predicted image is below a particular level of satisfaction, defined in terms of a threshold value (th), the forecasting process within each forecasting module is again triggered. Therefore, the weight values within each deep learning network and, hence, the predicted images from each forecasting module are tuned. The process continues until the threshold th is reached, or there is no improvement in the predicted image from the final forecasting module. In our proposed model, the *mean absolute error* (MAE) has been used as the measure of accuracy during the data/image tuning process.

Once the termination condition is reached, the dummy imagery $I_m, I_x, \ldots$ are considered as the predicted/reconstructed imagery of the corresponding time-series sequence.

III. Experimentation

This section provides the details of experimentation carried out on predicting missing images in a sequence of derived remote sensing imagery. The results of experimentation show a quite satisfactory performance of the proposed deep forecasting ensemble model (*Deep-STEP_FE*).

A. Dataset and Study Area

Experimentation has been carried out with a periodic series of NDVI imagery for eight consecutive time instants during 2004–2011, over the *Barddhaman* district, situated in the state of *West Bengal, India* (see Fig. 3). The primary source of these raster datasets is the Landsat-7 TM-5 satellite imagery from the *Land Process Distributed Active Archive Center* of the *United States Geological Survey* [13]. Later, ERDAS IMAGINE tool [15] has been utilized to generate the NDVI raster from the input raw satellite imagery. Two zones, containing several thousands of pixels, have been selected from the study area. The details of the study zones have been summarized in Table I. Moreover, the empirical study has been performed twice, considering that the NDVI images of year 2007 and year 2009 are missing. The years of missing images have been chosen randomly.

TABLE I
Details of Study Zones in the *Barddhaman* District

Zones	Bounding box details		
	Number of pixels	Top-Left	Bottom-Right
Zone-1	10 000	23.45 °N, 87.47 °E	23.42 °N, 87.50 °E
Zone-2	102 400	23.45 °N, 87.50 °E	23.37 °N, 87.59 °E

B. Experimental Setup

Experimentation with the proposed *Deep-STEP_FE* model and the other techniques has been performed using MATLAB 8.3.0.532 (R2014a) in Windows 2007 (64-bit OS, 3.10-GHz CPU, 4.00-GB RAM).

The performance of the proposed *Deep-STEP_FE* model has been evaluated in comparison with *six* other temporal and hybrid models, namely *nonlinear auto-regressive neural network* (NARNET, using NN toolbox of MATLAB), multilayer perceptron (MLP), *spatiotemporal ordinary kriging* (ST-OK, using "spacetime" package of R-tool), DSN [14], the *Deep-STEP_F* model, and the *Deep-STEP_B* model. The *Deep-STEP_F* is the Deep-STEP model [6], applied in the forward direction with the available data in earlier part (with respect to the missing image) of the time series, whereas the *Deep-STEP_B* is the Deep-STEP model [6], applied in the backward direction with the available data in subsequent part of the time series. The *Deep-STEP_B* model can also be treated as the backward forecasting model based on Deep-STEP. The combinations of training and test

TABLE II
SUMMARY OF TRAINING AND TEST DATA COMBINATIONS

Prediction Techniques	Prediction year 2007				Prediction year 2009			
	Training Phase		Test Phase		Training Phase		Test Phase	
	Feature set	Prediction	Feature set	Prediction	Feature set	Prediction	Feature set	Prediction
NARNET	2004–2005	2006	2005–2006	2007	2004–2007	2008	2005–2008	2009
MLP	2004–2005	2006	2005–2006	2007	2004–2007	2008	2005–2008	2009
DSN	2004–2005	2006	2005–2006	2007	2004–2007	2008	2005–2008	2009
Deep-STEP_B	2011–2009	2008	2010–2008	2007	2011	2010	2010	2009
Deep-STEP_F	2004–2005	2006	2005–2006	2007	2004–2007	2008	2005–2008	2009
Deep-STEP_FE	Module-1				Module-1			
	2004–2005	2006	2005–2006	2007	2004–2007	2008	2005–2008	2009
	Module-2				Module-2			
	2007(dummy)–2009	2010	2008–2010	2011	2009(dummy)	2010	2010	2011

TABLE III
GENERIC CONFIGURATION FOR THE DEEP LEARNING MODEL USED IN THE PROPOSED FORECASTING ENSEMBLE

Number of input units	Number of Hidden Layer	Number of units per hidden layer	Number of output units
Module-1 (bottom most): $(2s+1)^2$ (Effectively) Other modules: $(2s+1)^2 \times 2$	t (one per each module)	$(2s+1)^2$	Top most module:1 Other modules:$(2s+1)^2$
s = Spatial neighborhood coverage (see Section II-A); t = Number of training images; m = Module sequence number; $m = 1, 2, \ldots, t$			

data, as used by the various neural-network-based prediction models considered in our experimental study, are summarized in Table II in terms of the associated years.

1) Parameter Settings: It is stated in the work of Deng *et al.* [16], that in the case of DSN learning, the full-batch training gives a significantly lower error rate than all sizes of mini-batch. Since the proposed Deep-STEP_FE uses Deep-STEP, a variant of DSN learning, therefore, in our experimentation, we have used full batch for the training purpose. Furthermore, during gradient descent analysis, the number of epoch has been set to 200, the rate of learning has been fixed at 2, and the weights (w_{ij} and u_{ij}) in each of the learning modules in our deep architecture have been initialized randomly such that $w_{ij}, u_{ij} \in [-1, 1]$.

Now, apart from the parameters in gradient descent analysis, another key parameter corresponding to our deep learning model is the *spatial neighborhood coverage*, which defines the size of feature set and eventually determines the number of units in the input, hidden, and output layers of the deep network. In case the spatial neighborhood coverage is s, $(2s+1)^2$ numbers of pixels from the spatial neighborhood of previous time instant are considered to create the feature set for each target pixel in the bottom-most learning module. Moreover, the input feature set for any top-level module additionally contains $(2s+1)^2$ more elements, i.e., $(2 \times (2s+1)^2)$ in total, representing some new ST features learned from the lower level modules. Therefore, according to the module architecture, the number of input unit in the bottom-most level is $(2s+1)^2$ and that for any top module is $(2 \times (2s+1)^2)$. For the same reason, the number of hidden units at each module is set to $(2s+1)^2$. The detail of network configuration for the deep learning model used in the proposed forecasting ensemble is summarized in Table III.

C. Performance Metrics

The comparative performance study of the proposed forecasting ensemble model has been made with respect to four evaluation criteria: *root-mean-square error* (RMSE), MAE [17], *peak signal-to-noise ratio* (PSNR) [18], and *mean structural similarity* (MSSIM) index [19]. The RMSE and MAE have been used to estimate the overall loss in such a prediction process, whereas the PSNR and MSSIM have been used to assess the quality of the predicted NDVI imagery. The mathematical formulations for each of these metrics are given as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(V_o - V_p)^2} \tag{10}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|V_o - V_p| \tag{11}$$

where V_o and V_p denote the observed value and the corresponding predicted value of the variable (e.g., NDVI), respectively, and n is the total number of observations

$$\text{PSNR} = 20.\log_{10}\left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}}\right) \tag{12}$$

where MAX_I is the maximum possible pixel value in the image

$$\text{MSSIM}(R, X) = \frac{1}{N}\sum_{i=1}^{N}\text{SSIM}(r_i, x_i) \tag{13}$$

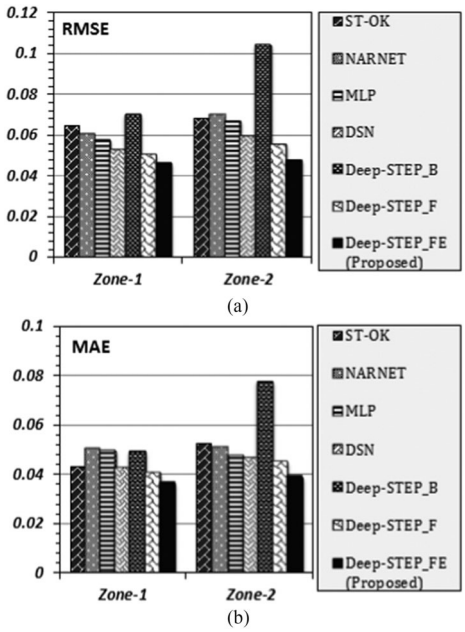

Fig. 4. Performance of missing NDVI imagery prediction for 2007. (a) RMSE. (b) MAE.

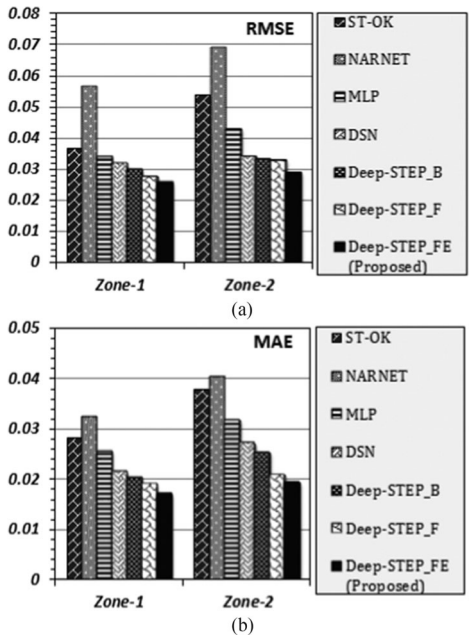

Fig. 5. Performance of missing NDVI imagery prediction for 2009. (a) RMSE. (b) MAE.

where R and X are the reference/original and the predicted images, respectively, SSIM is the *structural similarity* index [19], r_i and x_i are the image contents at the ith local window in the reference and the predicted images, respectively, and N is the number of local windows considered.

D. Results and Discussion

The accuracy of missing NDVI image prediction for the years 2007 and 2009 have been graphically plotted in terms of RMSE and MAE in Figs. 4 and 5, respectively, in comparison with various other prediction techniques. Figs. 6 and 7 depict the normalized error surfaces corresponding to the spatial distributions of NDVI for both the study zones. Furthermore, the predicted image quality assessment has been summarized in Tables IV and V for the years 2007 and 2009, respectively. By analyzing the different outcomes, as shown in Tables IV and V and Figs. 4–7, the following inferences can be drawn.

1) It is evident from Figs. 4 and 5 that the proposed deep forecasting ensemble (*Deep-STEP_FE*) outperforms the others producing least RMSE and MAE in predicting NDVI for both the study zones. The performance of *Deep-STEP_FE* is even better than that of a single Deep-STEP model [6], applied in the forward direction. This also indicates the effectiveness of the proposed deep-learning-based forecasting ensemble.
2) It can also be noted that the performance of backward forecasting (refer *Deep-STEP_B* in Fig. 4) is significantly poor compared to all the other prediction models applied in the forward direction. This supports the concept of causality constraint and strengthens the motivation of developing a forward forecasting ensemble, instead of using a forward–backward forecasting ensemble.
3) From Figs. 6 and 7, it may be observed that the deep-learning-based forecasting models (DSN, Deep-STEP_F, and Deep-STEP_FE) produce significantly lesser error distribution, compared to NARNET, MLP, and ST-OK. Furthermore, the proposed deep forecasting ensemble model (Deep-STEP_FE) is found to perform even better than DSN and single Deep-STEP, by generating the least error distributions over both the study zones. This proves the worth of considering deep network learning in prediction of missing data/imagery in multitemporal analysis.
4) The quality of the predicted imagery, as summarized in Tables IV and V in terms of PSNR and MSSIM, also reveals that the performance of the proposed forecasting ensemble (*Deep-STEP_FE*) is far better than that of the ST-OK, NARNET, MLP, DSN, and even stand-alone Deep-STEP model [6], for both Zone-1 and Zone-2.

Overall, the experimental study shows better performance of deep-learning-based approaches in predicting missing imagery in a time-series sequence. The study also demonstrates the effectiveness of the proposed deep forecasting ensemble (*Deep-STEP_FE*), compared to a single Deep-STEP forecasting model. Incidentally, the proposed ensemble model is not only applicable for reconstructing missing image at a single time instance. The generic architecture (see Section II) is applicable for predicting missing imagery from *multiple* a-periodic time instances as well.

1) Running Time: The execution time of the proposed Deep-STEP_FE model depends on the number of forecasting modules in the ensemble architecture and the number of learning modules within each forecasting module. However, in any case, the runtime complexity will be of the same order as that of the Deep-STEP [6], which shows comparative execution time, even slightly better than the original DSN model. Moreover, as per the overall architecture of the proposed forecasting ensemble, there remains huge scope to execute each of the constituting

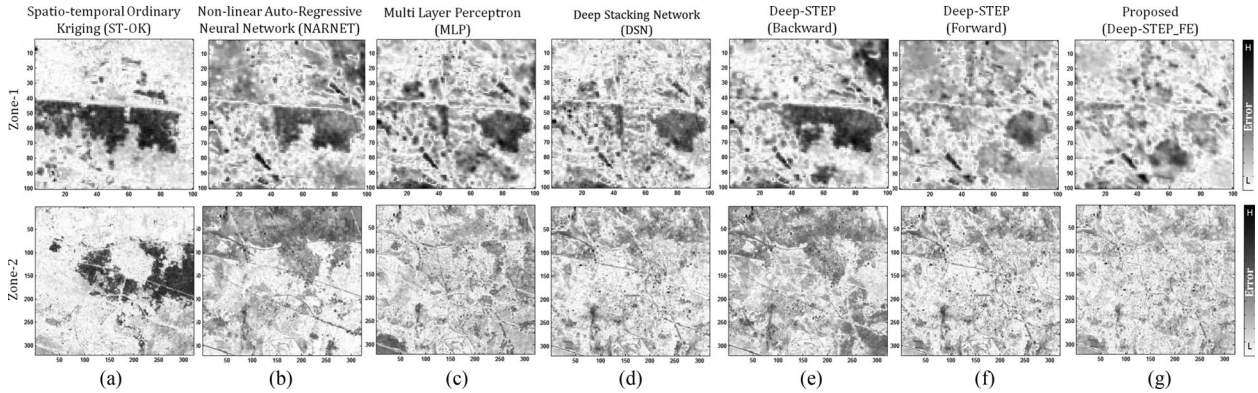

Fig. 6. (a)–(g) Normalized error surface corresponding to missing NDVI imagery prediction for 2007.

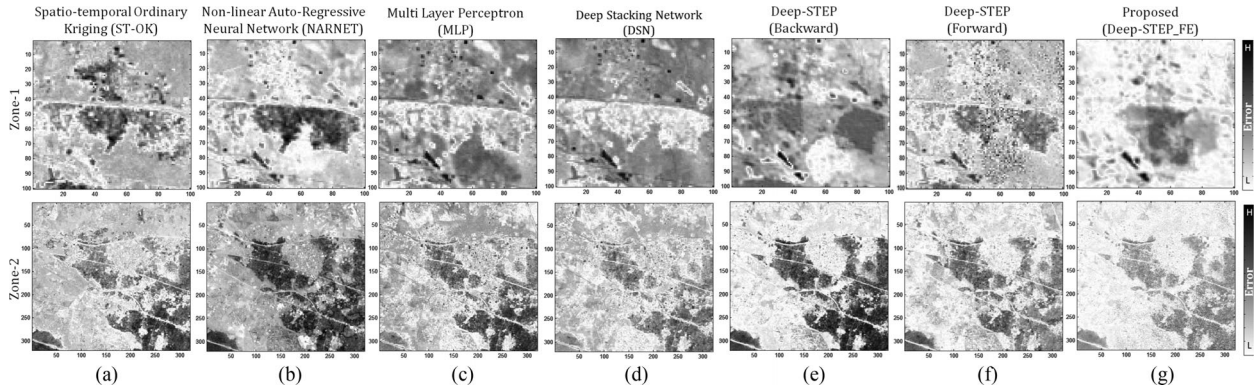

Fig. 7. (a)–(g) Normalized error surface corresponding to missing NDVI imagery prediction for 2009.

TABLE IV
COMPARATIVE STUDY OF THE PREDICTED IMAGE QUALITY FOR THE PROPOSED ENSEMBLE MODEL (DEEP-STEP_FE): YEAR 2007

Zones	Performance Metrics	ST-OK	NARNET	MLP	DSN	Deep-STEP_B	Deep-STEP_F	Deep-STEP_FE
Zone-1	PSNR (dB)	23.795	24.365	24.831	25.569	23.061	25.934	26.716
	MSSIM	0.7903	0.6425	0.6516	0.7436	0.6251	0.7931	0.8215
Zone-2	PSNR (dB)	23.291	23.111	23.513	24.537	19.670	25.119	26.438
	MSSIM	0.7747	0.6857	0.7624	0.7783	0.5049	0.7854	0.8167

TABLE V
COMPARATIVE STUDY OF THE PREDICTED IMAGE QUALITY FOR THE PROPOSED ENSEMBLE MODEL (DEEP-STEP_FE): YEAR 2009

Zones	Performance Metrics	ST-OK	NARNET	MLP	DSN	Deep-STEP_B	Deep-STEP_F	Deep-STEP_FE
Zone-1	PSNR (dB)	28.673	24.909	29.338	29.928	30.443	31.198	31.772
	MSSIM	0.7885	0.581	0.6676	0.8071	0.8357	0.8535	0.9136
Zone-2	PSNR (dB)	25.386	23.227	27.35	29.33	29.575	29.646	30.781
	MSSIM	0.7611	0.5157	0.6168	0.7258	0.7668	0.8066	0.8897

forecasting modules in parallel fashion and thereby to further reduce the running time.

2) Scalability Issue: The deep architecture used in our proposed forecasting ensemble (Deep-STEP_FE) is built in such a way that each of the learning modules in the architecture corresponds to a particular training year in sequence. Therefore, the total number of learning modules is equal to the total number of training images available. Now, as per our deep learning model, each of these learning modules is trained first to learn the spatial features at a particular time instant, and then, the modules are stacked on the top of each other to learn the complex temporal evolution pattern of these features. So, huge scope remains in

training the learning modules in CPU cluster, in parallel fashion, to handle the load of large-scale remote sensing data/imagery. Otherwise, if the proposed Deep-STEP_FE model is executed in a single stand-alone machine, then the size of the image should be restricted within a particular limit. In our present case study, we have used single stand-alone CPU and our model has been found to work well even with images containing one million pixels. In this respect, the proposed forecasting ensemble model (Deep-STEP_FE) can be considered to be fairly scalable. However, as already mentioned, in case the size of the image becomes too large, the approach may need to use CPU cluster, and in this regard, we have a plan for extending our approach to its parallel version for dealing with such a situation.

IV. Conclusion

The performance of remote sensing analyses is often severely deteriorated because of missing data, which could not be generated due to unavailability of source satellite imagery. This paper proposes a deep-learning-based model (*Deep-STEP_FE*) for predicting missing data in the remote sensing time series. The proposed prediction approach is based on Deep-STEP [6]. However, the novelty of the proposed approach is that, unlike Deep-STEP and other ST prediction models, the proposed ensemble model utilizes *both* the earlier and subsequent data in the remote sensing time series, without contradicting with the *causality constraint*. Consequently, this offers a useful way of utilizing available data/information in an effective manner. Experimentation has been carried out to predict missing NDVI imagery during the time period from 2004 to 2011 for *two* spatial zones in *Barddhaman, India*, comprising of several thousands of pixels. Comparative performance analysis with the state-of-the-art deep-learning-based ST prediction models (DSN, Deep-STEP, etc.) and conventional time-series prediction using ST-OK, MLP, and NARNET learning techniques demonstrates the superiority and effectiveness of the proposed deep forecasting ensemble model (*Deep-STEP_FE*) for missing data prediction. In future, the work can be extended to predict missing raw satellite imagery, consisting of multiple bands/layers of spectral information. The proposed model can also be upgraded to deal with very large scale remote sensing datasets by parallel execution of the constituting forecasting modules.

References

[1] B. P. Salmon, J. C. Olivier, K. J. Wessels, W. Kleynhans, F. Van den Bergh, and K. C. Steenkamp, "Unsupervised land cover change detection: Meaningful sequential time series analysis," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 2, pp. 327–335, Jun. 2011.

[2] A. Rahman, S. P. Aggarwal, M. Netzband, and S. Fazal, "Monitoring urban sprawl using remote sensing and GIS techniques of a fast growing urban centre, India," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 1, pp. 56–64, Mar. 2011.

[3] J. L. Crespo, M. Zorrilla, P. Bernardos, and E. Mora, "A new image prediction model based on spatio-temporal techniques," *Vis. Comput.*, vol. 23, no. 6, pp. 419–431, 2007.

[4] J. Snepvangers, G. Heuvelink, and J. Huisman, "Soil water content interpolation using spatio-temporal kriging with external drift," *Geoderma*, vol. 112, no. 3, pp. 253–271, 2003.

[5] T. A. Moahmed, N. El Gayar, and A. F. Atiya, "Forward and backward forecasting ensembles for the estimation of time series missing data," in *Proc. IAPR Workshop Artif. Neural Netw. Pattern Recognit.*, 2014, pp. 93–104.

[6] M. Das and S. K. Ghosh, "Deep-step: A deep learning approach for spatiotemporal prediction of remote sensing data," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1984–1988, Dec. 2016.

[7] H. Shen *et al.*, "Missing information reconstruction of remote sensing data: A technical review," *IEEE Geosci. Remote Sens. Mag.*, vol. 3, no. 3, pp. 61–85, Sep. 2015.

[8] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *J. Vis. Commun. Image Representation*, vol. 12, no. 4, pp. 436–449, 2001.

[9] N. Cressie and C. K. Wikle, *Statistics for Spatio-Temporal Data*. New York, NY, USA: Wiley, 2015.

[10] X. Li, H. Shen, L. Zhang, H. Zhang, and Q. Yuan, "Dead pixel completion of aqua modis band 6 using a robust m-estimator multiregression," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 4, pp. 768–772, Apr. 2014.

[11] X. Li, H. Shen, L. Zhang, H. Zhang, Q. Yuan, and G. Yang, "Recovering quantitative remote sensing products contaminated by thick clouds and shadows using multitemporal dictionary learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 7086–7098, Nov. 2014.

[12] S. Benabdelkader and F. Melgani, "Contextual spatiospectral postreconstruction of cloud-contaminated images," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 204–208, Apr. 2008.

[13] "USGS EarthExplorer: Land Processes Distributed Active Archive Center," 2014. [Online]. Available: https://lpdaac.usgs.gov/data_access/usgs_earthexplorer

[14] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3/4, pp. 197–387, 2014.

[15] "ERDAS IMAGINE: Hexagon Geospatial," 2014. [Online]. Available: http://www.hexagongeospatial.com/products/remote-sensing/erdas-imagine/overview

[16] L. Deng, B. Hutchinson, and D. Yu, "Parallel training for deep stacking networks," in *Proc. 13th Annu. Conf. Int. Speech Commun. Assoc.*, 2012, pp. 1–4.

[17] T.-H. Lee, "Loss functions in time series forecasting," *Int. Encyclopedia Soc. Sci.*, vol. 9, pp. 495–502, 2008.

[18] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

Monidipa Das (S'14) received the M.E. degree in computer science and engineering from the Indian Institute of Engineering Science and Technology, Shibpur, India, in 2013. She is currently working toward the Ph.D. degree with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India.

Her research interests include spatial and spatiotemporal data mining, soft computing, and machine learning.

Ms. Das is a member of the IEEE Computational Intelligence Society.

Soumya K. Ghosh (M'04) received the Ph.D. and M.Tech. degrees from the Department of Computer Science and Engineering, Indian Institute of Technology (IIT), Kharagpur, India. Presently, he is a Professor with the Department of Computer Science and Engineering, IIT Kharagpur. Before joining IIT Kharagpur, he worked for the Indian Space Research Organization in the area of satellite remote sensing and geographic information systems. He has more than 200 research papers in reputed journals and conference proceedings. His research interests include spatial data science, spatial web services, and cloud computing.

Dr. Ghosh is a member of the IEEE Geoscience and Remote Sensing Society.