SARDINE: A Self-Adaptive Recurrent Deep Incremental Network Model for Spatio-temporal Prediction of Remote Sensing Data

MONIDIPA DAS, Nanyang Technological University, Singapore
MAHARDHIKA PRATAMA, Nanyang Technological University, Singapore
SOUMYA K. GHOSH, Indian Institute of Technology Kharagpur, India

Timely and accurate prediction of remote sensing data is of utmost importance especially in a situation where the predicted data is utilized to provide insights into emerging issues, like environmental nowcasting. Significant research progress can be found to date in devising variants of neural network (NN) models to fulfil this requirement by improving feature extraction and dynamic process representation power. Nevertheless, all these existing NN models are built upon rigid structures which often fail to maintain trade-off between bias and variance, and consequently, need to spend a lot of time to empirically determine the most appropriate network configuration. This paper proposes SARDINE, a novel variant of deep recurrent neural network with intrinsic capability of self-constructing the network structure in dynamic and incremental fashion while learning from observed data samples. Moreover, the proposed SARDINE is able to model the spatial feature evolution while scanning the data in single pass manner and this further saves significant time when dealing with remote sensing imagery containing millions of pixels. Subsequently, we employ SARDINE in combination with a spatial influence mapping unit to accomplish the prediction. The effectiveness of the proposed model is evaluated in terms of predicting time series of normalized difference vegetation index (NDVI) data derived from Landsat TM-5 and MODIS Terra satellite imagery. The experimental result demonstrates that the SARDINE-based prediction is able to achieve state-of-the-art accuracy with significantly reduced computational cost.

CCS Concepts: • **Information systems** → **Spatio-temporal systems**; *Geographic information systems*.

Additional Key Words and Phrases: Spatio-temporal prediction, Evolving RNN, Incremental model, Remote sensing

ACM Reference Format:
Monidipa Das, Mahardhika Pratama, and Soumya K. Ghosh. 2020. SARDINE: A Self-Adaptive Recurrent Deep Incremental Network Model for Spatio-temporal Prediction of Remote Sensing Data. *ACM Trans. Spatial Algorithms Syst.* XX, X, Article XXX (January 2020), 28 pages. https://doi.org/10.1145/1122445.1122456

Authors' addresses: Monidipa Das, Nanyang Technological University, Singapore, monidipadas@ntu.edu.sg; Mahardhika Pratama, Nanyang Technological University, Singapore, mpratama@ntu.edu.sg; Soumya K. Ghosh, Indian Institute of Technology Kharagpur, India, skg@cse.iitkgp.ac.in.

2374-0353/2020/1-ARTXXX $15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In the present era of advanced science and technology, the remote sensing data play central role in studying various spatio-temporal processes on the earth surface. Whether captured by a geo-stationary or polar orbiting satellite, the remotely sensed imagery provide continuous and synoptic observations covering large areas of the earth surface, which are enormously useful for detailed monitoring and damage assessment purpose. Analysis of urban sprawl pattern [12, 30], assessment of coastal vulnerability [6], monitoring natural disasters (like flood, cyclone etc.) [33] are popular domains with wide applicability of remote sensing data.

However, analyzing remote sensing data often becomes challenging due to missing image or missing values introduced by defective sensor, low temporal frequency, cloud cover, and other poor conditions of the atmosphere [4][11]. Consequently, the usability of the data becomes limited in many situations. In order to deal with this issue, low-quality or defected data are often discarded and the missing data are reconstructed by means of prediction from observed data. The gapfill-MAP [19], spatio-temporal interpolation techniques [8], TIMESAT [19] etc. are some popular approaches in this respect. Research efforts can also be noticed in employing techniques like multi-sensor data fusion [37] and multi-temporal-complementation [26] for this purpose. The situation becomes even more serious when all the complementary spatial information for a particular time instance is missing in a series of derived remotely sensed imagery, and in such case, it becomes necessary to predict the entire image by utilizing the spatio-temporal information from the available data. Recently, the Deep-STEP [10] and the STBN [13] models are typically proposed to deal with such scenario. This paper also focuses on spatio-temporal prediction of full derived image for a particular time stamp, based on the observed data from earlier time instances, while assuming all the source imagery are captured through similar sensor.

1.1 Challenges and Motivations

With the recent advancements of AI and machine learning, there has been significant progress in remote sensing data prediction technology. More prominently, variants of neural network models with their deep architectures and hierarchical feature learning capability have been found to show promising performance in spatio-temporal prediction of satellite remote sensing imagery. However, despite these progressions, there remain several issues such as fixed network architecture, multi-pass parameter learning, and large parameter requirements, which impose serious challenges especially when the prediction is necessary to be accomplished in a timely manner. A timely as well as accurate prediction of remote sensing data plays significant role especially in a situation when the predicted data is utilized to provide insights into urgent issues, like environmental nowcasting, civil protection alarming etc. Though variants of neural-network-based models in the literature [40][32] have already demonstrated their effectiveness in remote sensing image prediction, the successful applications of these models are often restricted because of the following concerns.

- **Fixed network architecture:** The existing models mostly aim at improvising feature extraction and dynamic process representation power, and pay little focus on the issue of on-the-fly structural adaptation of the model. Because of rigid network structure, these models often fail to maintain trade off between bias and variance. That means, either the model can move into over-fitting zone due to too complex network architecture and show high degree of variance, or it can move into under-fitting zone due to insufficient network configuration and show high degree of bias [1]. Further, because of rigid structure of the network, these models often fail to appropriately deal with the change

in spatial contexts throughout the image. In order to avoid such situations, the model needs to empirically determine the most appropriate network configuration for each different scenario. This requires substantial time to be spent, and as a consequence, the model becomes unsuitable for timely prediction in diverse contexts.

- **Multi-pass parameter learning:** Another major limitation with several of the existing approaches lies in their parameter updating process which often involves multi-pass data scanning strategy i.e. iterating over the training data. However, in case the prediction is meant for quick decision making process, the multi-pass parameter learning approach may obstruct the model to behave desirably, since each remote sensing image contains very large number of pixels.
- **Large parameter requirements:** Although the various deep learning models like convolutional neural networks are found to work extremely well with image data, in many of the cases, these require very large number of parameters that grow substantially with the increasing volume of the data. Similar issues also arise with the long-short term memory (LSTM) recurrent neural network models. Though LSTMs perform notably well in predicting image sequences, the success of these models are mainly due to various gating mechanisms which need to exploit a large number of parameters, and this eventually leads to high computational cost.

Motivated by the above-mentioned issues, in this paper we attempt to develop a neural network variant with flexible network architecture and single-pass learning ability, so as to accomplish spatio-temporal prediction of derived remote sensing imagery in a timely manner.

1.2 Our Contributions

This paper proposes SARDINE, a novel variant of recurrent neural network with the intrinsic capability of self constructing the network structure in dynamic and incremental fashion while learning from observed data samples. Moreover, the proposed SARDINE is able to model the spatial feature evolution while scanning the data in single pass manner and this further saves significant amount of time during the course of prediction. The effectiveness of the proposed prediction model is evaluated in terms of predicting time series of remotely sensed *normalized difference vegetation index* (NDVI) data derived from Landsat TM-5 and MODIS Terra satellite imagery [35]. The comparative study is carried out with a number of *traditional baselines* as well as *state-of-the-art deep learning techniques* for remote sensing image prediction. Our key contributions in this work are as follows:

- We propose SARDINE as a deep recurrent neural network variant for learning the temporal evolution of spatial features in *single pass, on-the-fly*, and *incremental manner*.
- The proposed SARDINE features flexible architecture and *self-evolution power*. Accordingly, it begins the learning using a single hidden unit in a single hidden layer and auto-construct the network to cope with variants of spatial contexts of the pixels in the imagery.
- The network structure of SARDINE is *auto-adjusted* and needs no additional empirical study to find the best configuration. This saves substantial amount of computational time and also helps *optimal exploitation of network parameters*.
- The proposed SARDINE is designed to learn using *teacher forcing* mechanism, and thus, overcomes the computational burden and gradient vanishing/exploding problems in conventional RNN learning that uses back propagation through time (BPTT).

- Finally, we develop SARDINE-based *prediction model* with embedded mechanism of auto-modeling spatial influence, for spatio-temporal prediction of derived remote sensing imagery;

To the best of authors' knowledge, this is the *first work* on a self-evolving *deep* architecture for a recurrent network model and employing the same in remote sensing image prediction. The incremental learning capability of the proposed model offers *added benefit* while dealing with variants of spatial contexts from widely spread region as captured through the satellite imagery. To be noted, though our earlier work in [14] introduced the concept of evolving recurrent neural network, the model is not self-evolving in terms of *deep* architecture. Further, since the model in [14] uses only the norm of previous output without any explicit recurrent connection, the model [14] loses significant information from the past and becomes unsuitable for the present image regression context.

The rest of the paper is organized in following manner. The Section 2 discusses on the existing techniques for predicting remote sensing data. A formal description of the problem formulation is presented in Section 3. The detailed illustration of the methodology is provided in Section 4 with a special focus on proposed self-adaptive recurrent deep incremental network model, SARDINE. The experimental evaluation of the proposed model is extensively described in Section 5 in comparison with a number of baseline techniques. Finally, the concluding remarks along with several future prospects are summarized in Section 6.

2 RELATED WORKS ON PREDICTION OF REMOTE SENSING DATA

Prediction under spatio-temporal framework is one of the hot research topics in recent years and has attracted considerable attentions from diverse areas [22, 25, 42]. The domain of satellite remote sensing is also not an exception in this respect. However, the existing spatio-temporal prediction models in this domain are mostly intended for remote sensing image classification, while the regression of remote sensing data derived from satellite imagery still remains comparatively little explored. Nevertheless, as per the current context of this paper, we will focus on the space-time regression of single-sensor satellite remote sensing data, with typical distinctions between conventional statistical and deep learning models.

2.1 Conventional models

Among the various conventional statistical approaches, the space-time Kriging [8], singular spectrum analysis, linear regression [24], spatio-temporally weighted regression [4], Kalman filtering [31], and the spatiotemporal Markov random field models [5] are widely used in remote sensing image prediction. However, one of the key limitations of these techniques is that these are mostly not scalable to large datasets and also sometimes not suitable for parallelization. Further, these approaches are more applicable on predicting corrupted/missing pixel values in the imagery and often fail in the situation when missing values fall within large gaps. For the same reason, these are not at all suitable for forecasting the full imagery for some future time stamps. Another major issue with these models is that these lack proper quantification of prediction uncertainties and also suffer from low speed due to computationally expensive methods requiring large-scale storage. Though the recently proposed spatio-temporal Bayesian network (STBN)-based prediction model [13] is found to show some advantages regarding uncertainty handling, large-scale data modeling, computational resource requirements etc., like all other previously-mentioned models, STBN is also based on *hand-crafted architecture* for learning the space-time dependency.

2.2 Deep learning models

The key advantage of the deep learning approaches remain in their automatic and hierarchical feature extraction capability from huge volume of datasets, which eventually makes them highly potential to deal with satellite remote sensing images, each of which contains very large number of pixels. Among the various deep learning approaches, Deep-STEP [10] and Deep-STEP_FE [11] are some recently proposed and pioneering models which are able to predict missing derived remote sensing imagery for future or intermediate time stamps. Both Deep-STEP and Deep-STEP_FE are derived from deep stacking network (DSN) [15] which is based on the concept of stacking simple modules built upon single hidden-layer network. Currently, Marzano [27] have used RC-NN as a recurrent neural network (RNN) variant which is found to be effective for nowcasting rainfall from Multisatellite passive-sensor images. Zhang et al. [46] have proposed a recurrent neural network model with long short term memory (LSTM) architecture to predict sea surface temperature from high resolution remote sensing data. There also exist some works employing convolutional neural network (CNN) for space-time regression from satellite remote sensing imagery [23]. However, these are mostly focused towards mapping of remote sensing data into a different quantitative variable to be predicted, instead of predicting the imagery over the original data. Recent research trend also shows an attempt to use combination of RNN-LSTM and convolutional neural network to utilize the power of LSTM for handling sequential input imagery and that of CNN for better extraction of spatial/spatio-temporal features. For example, You et al. [44] have applied combined LSTM and CNN with incorporated Gaussian process for crop yield prediction. Yang et al. [41] have used LSTM model with added convolutional layer for predicting high resolution sea surface temperature data derived from satellite remote sensing imagery. However, the use of large number of parameters becomes a major issue which restricts these models to successfully operate only when the number of image in the sequence is quite large. Further, all these models are built upon fixed-layered network architecture, and thus, can contribute less in the scenario of prompt prediction, as thoroughly discussed in the Section 1.1.

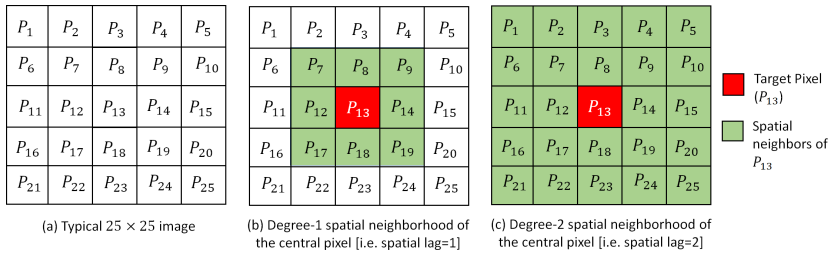
In order to address the issues with rigid architecture, iterative learning, and large parameter requirements, in this paper we have proposed a new variant of deep recurrent neural network with online learning capability and self-evolution property, which subsequently leads the model to deal with optimal number of parameters. When evaluated on remote sensing time series of NDVI (normalized difference vegetation index) imagery, the proposed model consistently outperforms the state-of-the-art deep-learning models.

3 PROBLEM FORMULATION

In this section, we present the overall spatio-temporal prediction problem as the prediction of derived remote sensing imagery with consideration to the influence of temporally evolving spatial features in the neighborhood of each pixel.

Let's denote the derived remote sensing imagery at time instance t as I_t and any pixel corresponding to the spatial location (p, q) in I_t as $P(p, q, t)$, where $P(p, q, t) \in \mathbb{R}$. Then, given the sequence of remote sensing imagery $I = \{I_1, I_2, \cdots, I_{T-1}, I_T\}$ over any derived variable for T number of time stamps, the goal is to predict I_{T+1}, considering each $P(p, q, T + 1) \in I_{T+1}$ to be attributed by the spatio-temporal features from a predefined spatial neighborhood coverage S. The concept of spatial neighborhood coverage is depicted in the Fig. 1. As illustrated through the figure, for any pixel $P(p, q, t)$, the spatial neighborhood coverage S of degree d is comprised of $\left[(2d + 1)^2 - 1\right]$ number of neighboring pixels which

can be represented as $S(P) = S(P(p,q,t)) = \{P(p \pm 1, q \pm 1, t), \cdots, P(p \pm d, q \pm d, t)\} - \{P(p,q,t)\}$.



(a) Typical 25×25 image (b) Degree-1 spatial neighborhood of the central pixel [i.e. spatial lag=1] (c) Degree-2 spatial neighborhood of the central pixel [i.e. spatial lag=2]

Fig. 1. Illustration of spatial neighborhood coverage with respect to a central pixel

Under this scenario, the generic objective function can be formulated as follows:

$$\min_{\widehat{I}_{T+1} \in \mathbb{R}^N, I_{T+1} \in \mathbb{R}^N} \quad loss\left(I_{T+1}, \widehat{I}_{T+1}\right) \tag{1}$$

where,

$$\widehat{I}_{T+1} = \underset{I_{T+1} \in \mathbb{R}^N}{\operatorname{argmax}} Pr(I_{T+1} | I_1, I_2, \cdots, I_{T-1}, I_T) \tag{2}$$

N is the total number of pixels in each image and *loss* indicates the loss function. This can also be represented approximately as follows.

$$\min_{\widehat{P}(p,q,T+1) \in \mathbb{R}, P(p,q,T+1) \in \mathbb{R}} \quad \sum_{i,j=1}^{\sqrt{N}} loss\left(P(p_i, q_j, T+1), \widehat{P}(p_i, q_j, T+1)\right) \tag{3}$$

where,

$$\widehat{P}(p,q,T+1) = \underset{P(p,q,T+1) \in \mathbb{R}}{\operatorname{argmax}} Pr(P(p,q,T+1) | P(p \pm d, q \pm d, 1), \cdots P(p \pm d, q \pm d, T)) \tag{4}$$

This paper attempts to solve the objective function by employing a deep evolving recurrent network model, along with an embedded spatial influence mapping unit working in parallel to capture the spatial influence. The detailed architecture as well as working principle of the proposed model is described in the following section.

4 PROPOSED PREDICTION MODEL BASED ON SARDINE

4.1 Model Overview

The overall flow of the proposed prediction model is depicted in the Fig. 2. At first, the model learns how the neighboring spatial features of each pixel evolves with time, and eventually, predicts the condition of each pixel of the prediction image, based on the predicted condition of the neighboring pixels.

It may be observed from the Fig. 2 that, the proposed prediction model is comprised of three major modules corresponding to *spatial feature representation*, modeling of *spatial feature evolution*, and finally, *prediction with consideration to the spatial influence*. The spatial feature representation (Module-I) as well as spatial influence modeling (Module-III) are based on the assumption that a pixel at a particular location is likely to have similar value as that of the pixels in its spatial/temporal neighborhood. This is supported by Tobler's

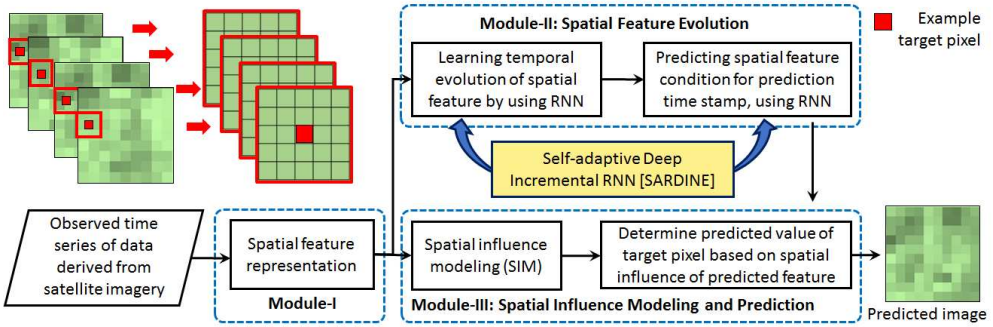


Fig. 2. Proposed prediction model based on SARDINE

'First Law of Geography' [3]. Accordingly, each pixel of the forecast image is finally produced as a function of predicted values of the neighboring pixels which are, meanwhile, determined as per the process of spatial feature evolution, captured through the Module-II. Incidentally, the modeling of spatial feature evolution is itself a crucial task as it may be driven by several geo-spatial or anthropogenic process intervention in the background. In our model, we employ a deep recurrent neural network model (SARDINE) for this purpose. The idea is to indirectly capture the effect of the background processes in terms of the temporal change in surrounding spatial features represented at various levels of abstractions. The learning of multi-level feature abstraction is accomplished through the incremental deep architecture of SARDINE, whereas the modeling of temporal change in the feature is achieved through its self-evolving recurrent structure. The dynamically adaptive network structure of SARDINE not only helps to auto-adjust the model capability with the change in spatial context, but also this resolves the issue of timely prediction as it does not require additional empirical analysis to identify the best suitable configuration.

The technical details for each key module in our proposed model are provided in the subsequent subsections.

4.2 Module-I: Input-Output Spatial Feature Representation

The primary aim of this module is to represent each pixel in terms of intensity of its neighboring pixels so that these form the input spatial feature vector for the corresponding pixel and help in defining the associated record in the dataset. This makes the proposed model consistent with both our problem statement and the state-of-the-art convention [7, 10]. Numerically, this can be expressed as follows:

$$P(p, q, t) = \psi(P(p \pm d, q \pm d, t)) \tag{5}$$

where, $d \neq 0$ denotes the degree of spatial neighborhood coverage in space. Thus, the input feature for a data record corresponding to $P(p, q, t)$ becomes: $\langle P(p-d, q-d, t)$, $P(p-d+1, q-d+1, t), \cdots P(p+d-1, q+d-1, t), P(p+d, q+d, t)\rangle$.

Now, the output feature for each data record is prepared in two ways to serve the objectives of Module-II and Module-III respectively. For the Module-II, the output feature set is prepared based on the spatial feature of the corresponding pixel in the immediate next time stamp. Thus, for Module-II, the output feature set for a data record corresponding to $P(p, q, t)$ is represented in terms of neighboring pixel intensity in the next time stamp as follows: $\langle P(p-d, q-d, t+1), P(p-d+1, q-d+1, t+1), \cdots P(p+d-1, q+d-1, t+1)$,

$P(p+d, q+d, t+1)\rangle$. Accordingly, a dataset of dimension $[N \times (2M)]$ is produced for the Module-II, with consideration to each individual timestamp t, where $M = (2d+1)^2 - 1$ is the size of input/output spatial feature vector associated with each of the N pixels observed at t.

On the other side, the output feature set to be used in Module-III is prepared to reflect the condition of each pixel in central position with respect to its spatial neighbors. Thus, for Module-III, the output feature set in a data record corresponding to a pixel at (p, q) position is represented in terms of its own pixel intensity as follows: $\langle P(p, q, t)\rangle$. On following the same way, a dataset of dimension $[N \times (M+1)]$ is produced for the Module-III, with consideration to each individual timestamp t, where $M = (2d+1)^2 - 1$ is the size of input spatial feature vector associated with each of the N pixels observed at t.

4.3 Module-II: Modeling of Spatial Feature Evolution using SARDINE

The key objective of this module is to hierarchically learn complex rules/functions for modeling the temporal evolution of spatial features during the observed time period. Later, in Module-III, the spatial influence of these temporally evolved features are utilized to predict each pixel value in the prediction image (refer to Fig. 4). The modeling of Spatial feature evolution in the Module-II is accomplished by a self-adaptive variant of recurrent neural network, termed as SARDINE, which is one of the key contributions of this paper. The details of recurrent architecture, parameter learning policy, and layer adaptation mechanisms of proposed SARDINE is thoroughly described in the subsequent subsections.

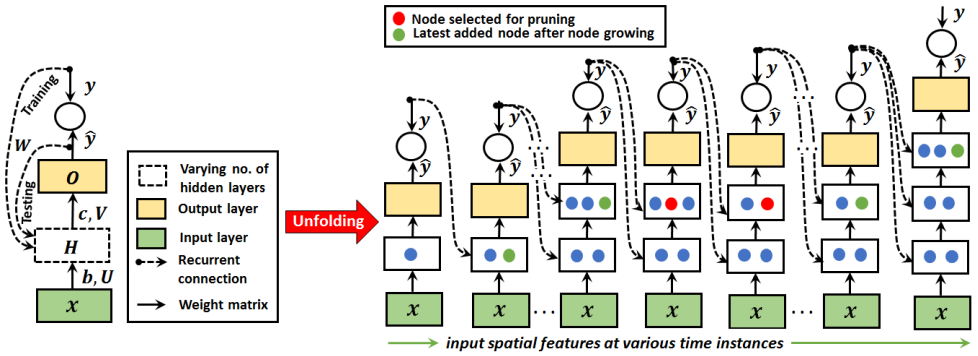


Fig. 3. Recurrent architecture of SARDINE and a typical state of its unfolded computational graph at a particular instant

4.3.1 Architecture. The recurrent network architecture along with the associated unfolded computational graph of our proposed SARDINE is depicted in Fig. 3. As shown in the figure, at each time instance t, the input to SARDINE is $x^{(t)\top} \in \mathbb{R}^M$ ($M = \left[(2d+1)^2 - 1\right]$ is the dimension of input feature), the hidden layer activation is $H_{(k)}^{(t)\top} \in \mathbb{R}^{e_k}$ (e_k is hidden unit count for layer k, $1 \leq k \leq l$), un-stretched output $O^{(t)}$ is subsequently updated using *linear stretching* function to generate the predicted output $\widehat{y}^{(t)}$ in the desired range corresponding to the associated variable. Typically, for SARDINE, $\widehat{y}^{(t)\top} \in \mathbb{R}^M$ (where M is the number of spatial features from neighborhood) and the loss is $L^{(t)}$. The connections between input $x^{(t)}$ and the first hidden layer $H_{(1)}^{(t)}$) are parameterized by weight matrix $\mathbf{U}_{[e_1 \times M]} = \mathbf{V}_{(\mathbf{0})_{[e_1 \times M]}}$;

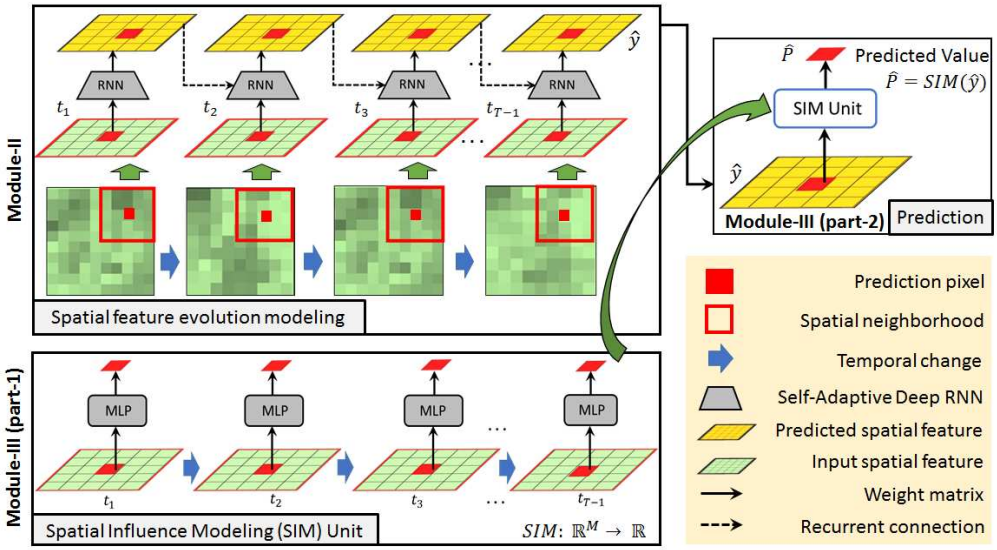


Fig. 4. Typical architecture of learning within the Module-II and Module-III of the proposed prediction model

the feed-forward connections between each k-th and $(k+1)$-th hidden layer are parameterized with weight matrix $\mathbf{V}_{(\mathbf{k})[e_{k+1} \times e_k]}$ where $(1 \leq k < l)$; and the forward connections between the top-most hidden layer $H_{(l)}^{(t)}$ and the output layer $O^{(t)}$ are parameterized with $\mathbf{V}_{(\mathbf{k})[M \times e_k]}$, where $k = l$. Most importantly, neither the e_k $(1 \leq k \leq l)$ nor the l is fixed for SARDINE. Both change dynamically depending on the spatial location of the input pixel during the training time. Moreover, in SARDINE, the recurrent connections exist between output and hidden layers, as represented using dashed arrows in the Fig. 3 and are parameterized with weight matrix $\mathbf{W}_{(\mathbf{k})[e_k \times M]}$ where $(1 \leq k \leq l)$. Because of these output-to hidden recurrent connections, the proposed SARDINE can learn from exact outputs from previous time stamps according to the *teacher forcing policy* [21] and thereby ensures greater parallelization in the training phase.

4.3.2 Parameter Learning. This section provides a detailed discussion on parameter learning of SARDINE in terms of forward propagation and backward propagation computation.

Forward-propagation computation. The forward propagation in SARDINE is performed by applying ReLU (rectified linear unit) and linear stretching function in the hidden layer and output layer, respectively. Accordingly, the hidden layer activation (H) computation becomes as follows.

When $k = 1$,

$$H_{(k)}^{i(t)} = max\left(0, \left[b_{(k)}^{i} + U_i x^{(t)} + W_{(k)i} y_{(t-1)}\right]\right) \tag{6}$$

and when $k > 1$

$$H_{(k)}^{i(t)} = max\left(0, \left[b_{(k)}^{i} + V_{(k-1)i} H_{(k-1)}^{(t)} + W_{(k)i} y_{(t-1)}\right]\right) \tag{7}$$

Here, $i = 1, 2, \cdots e_k$ denotes the hidden unit in the k-th hidden layer $H_{(k)}^{(t)}$ at time t, ${b_{(k)}}^{\top} \in \mathbb{R}^{e_k}$ denotes the bias for k-th level. At the time of testing, the actual output $y^{(t-1)}$ is replaced with the predicted output $\widehat{y}^{(t-1)}$, as depicted in Fig. 3. The use of ReLU as the hidden layer activation ensures that the model becomes free from vanishing gradient issue even when the depth of the network $(l+1)$ is increased. To be noted, in the Eq. 7, the concept of using the output from previous layer and the original feature from earlier timestamp is different from that proposed for deep residual neural network [22]. Our Eq. 7 is formulated to realize the *recurrent connection* as influenced from the *teacher forcing principle* [21], whereas the formulation proposed in [22] is primarily designed to realize "shortcut connections" in feed-forward neural networks.

Once the hidden layer activations are determined, the un-stretched output at t is estimated as follows:

$$O^{(t)} = c + V_{(k)} H_{(k)}^{(t)} \tag{8}$$

where $k = l$, and $c^{\top} \in \mathbb{R}^{M}$ denotes the bias in the output layer. The un-stretched outcomes are obtained within a range $[r_1, r_2]$ which needs to be further stretched into the desired range $[s_1, s_2]$ to generate the predicted output $\widehat{y}^{(t)}$ in following manner:

$$\widehat{y}^{(t)} = g(O^{(t)}) = aO^{(t)} + z \tag{9}$$

where, g is the linear stretching function [20], $a = \frac{(s_2 - s_1)}{(r_2 - r_1)}$, and $z = \left(s_1 - \frac{r_1 \times (s_2 - s_1)}{(r_2 - r_1)}\right)$.

After the estimation of predicted output $\widehat{y}$, the squared loss L is computed as follows:

$$L(y, \widehat{y}) = \frac{1}{2M} \sum_{i=1}^{M} (y_i - \widehat{y}_i)^2 \tag{10}$$

Here y denotes the observed value at the same time stamp. As per the problem definition (refer to Section 3), the predicted value $\widehat{y}$ here is nothing but the predicted spatial features $\widehat{S}(P) = \widehat{S}(p, q, t+1)$ for the pixel $P(p, q, t+1)$.

Back-propagation computation. The back-propagation computation in SARDINE is performed by applying stochastic gradient descent technique on the unfolded computational graph. However, since the recurrent connection in SARDINE is achieved through output-to-hidden connection, according to the *teacher forcing principle* [21], the network can directly use the spatial feature from earlier time stamps rather than using the predicted outputs. Thus, the back-propagation algorithm can be applied in isolation to each time stamp, instead of using generalized back-propagation-through-time (BPTT) algorithm. This ultimately reduces the computational burden and also resolves the exploding-gradient/vanishing-gradient problem, as often faced by traditional RNN with hidden-to-hidden recurrent connection [21]. Hence, the gradient computation in SARDINE for each time stamp becomes as follows.

The recursive computation is initiated with $\frac{\partial L}{\partial L^{(t)}} = 1$ and then it proceeds to the output layer and hidden layers.

$$\frac{\partial L}{\partial \widehat{y}_i^{(t)}} = \left(\nabla_{\widehat{y}^{(t)}} L\right)_i = -\left(y_i^{(t)} - \widehat{y}_i^{(t)}\right) \tag{11}$$

$$\frac{\partial L}{\partial O_i^{(t)}} = \left(\nabla_{O^{(t)}} L\right)_i = -a\left(y_i^{(t)} - \widehat{y}_i^{(t)}\right) \tag{12}$$

When $k = l$,

$$\nabla_{H_{(k)}^{(t)}} L = V_{(k)}^{\top} \left(\nabla_{O^{(t)}} L\right) \tag{13}$$

and when $1 \leq k < l$,

$$\begin{aligned} \nabla_{H_{(k)}^{(t)}} L &= \left(\nabla_{H_{(k+1)}^{(t)}} L\right), \text{ if } H_{(k+1)}^{(t)} > 0 \\ &= 0, \text{ otherwise} \end{aligned} \tag{14}$$

After the gradient computation for the internal nodes is over, the gradients for the parameters are estimated as follows:

Gradient calculation for bias parameters:

$$\nabla_c L = \left(\frac{\partial O^{(t)}}{\partial c}\right)^{\top} \left(\nabla_{O^{(t)}} L\right) = \left(\nabla_{O^{(t)}} L\right) \tag{15}$$

$$\begin{aligned} \nabla_{b_{(k)}} L &= \left(\nabla_{H_{(k)}^{(t)}} L\right), \text{ if } H_{(k)}^{(t)} > 0 \\ &= 0, \text{ otherwise} \end{aligned} \tag{16}$$

where $1 \leq k \leq l$.

Gradient calculation for weight parameters:

$$\nabla_{V_{(k)}} L = \left(\nabla_{O^{(t)}} L\right) \cdot \left(H_{(k)}^{(t)}\right)^{\top} \tag{17}$$

when $k = l$, and

$$\begin{aligned} \nabla_{V_{(k)}} L &= \left(\nabla_{H_{(k+1)}^{(t)}} L\right) \left(H_{(k)}^{(t)}\right)^{\top}, \text{ if } H_{(k+1)}^{(t)} > 0 \\ &= 0, \text{ otherwise} \end{aligned} \tag{18}$$

when $1 \leq k < l$.

$$\begin{aligned} \nabla_{W_{(k)}} L &= \left(\nabla_{H_{(k)}^{(t)}} L\right) \left(y^{(t-1)}\right)^{\top}, \text{ if } H_{(k)}^{(t)} > 0 \\ &= 0, \text{ otherwise} \end{aligned} \tag{19}$$

when $1 \leq k \leq l$.

4.3.3 Self-adjustment of within layer architecture. This section describes the self-evolution principle of SARDINE in adapting the within layer structure. The overall formulation for layer structure adaptation as adopted by proposed SARDINE is derived based on *network significance* (NS) method [1, 14], which is defined on the concept of expectation of mean squared error in prediction. Formally, this is expressed in terms of bias and variance formulation as given below.

$$NS = Bias(\widehat{y})^2 + Var(\widehat{y}) \tag{20}$$

The key utility of NS method lies in its capability of estimating the predictive model quality by direct assessment of the possible over-fitting/under-fitting condition of the model, which is not normally captured through standard system error index. A high NS value signals either a high variance condition, indicating model overfitting issue, or signals a high bias condition, indicating model underfitting issue.

For the proposed SARDINE, the bias-variance formulation is derived based on Eq. 20 in following manner.

$$NS = (E[\widehat{y}^2] - E[\widehat{y}]^2) + (E[\widehat{y}] - y)^2 \tag{21}$$

where, $E[\widehat{y}]$ represents the expectation of output from SARDINE, and it can be recursively estimated for any time instance t as follows.

$$E[\widehat{y}] = \int_{-\infty}^{\infty} (a \cdot O + z) p(O) dO = a \cdot E[O] + z \tag{22}$$

$$E[O] = \int_{-\infty}^{\infty} (c + V \cdot H) p(H) dH = c + V \cdot E[H] \tag{23}$$

$$E[H_{(k)}] = \int_{-\infty}^{\infty} max(0, H_{(k-1)}) p(H_{(k-1)}) dH_{(k-1)} \tag{24}$$

Now, as per the theory of *spatial autocorrelation* [3, 9], the spatial features from the neighboring locations are more likely to be similar. Accordingly, we can assume that the input spatial feature shows normal distribution with low standard deviation, and so, the value of $E[H_{(k)}]$ when $k = 1$ can be approximated as follows, where μ indicates the mean of feature values at the given time stamp.

$$E[H_{(1)}] = b_{(1)} + \left(U + W_{(1)}\right) \mu \tag{25}$$

Subsequently, we can estimate $E[H_{(k)}]$ (when $k > 1$), in recursive manner, using the Eq. 24. In a similar fashion, we can recursively calculate $E[\widehat{y}^2]$, considering the approximate value of $E[H_{(1)}]$ as $b_{(1)} + \left(U + W_{(1)}\right) \mu^2$.

After the NS is estimated, it is used to update the structural configuration of hidden layer in SARDINE, in a manner as illustrated in the subsequent parts of this section.

Hidden unit growing mechanism. The purpose of hidden unit growing is to overcome the under-fitting situation which is attributed by high bias condition. This is addressed by increasing the structural complexity of the network, i.e., by adding more nodes/units in the hidden layer. Thus the high bias condition for adding new hidden unit can be mathematically formulated in following manner:

$$\mu^t_{Bias} + \sigma^t_{Bias} \geq \mu^{min}_{Bias} + \pi \sigma^{min}_{Bias} \tag{26}$$

Here μ^t_{Bias} and σ^t_{Bias} respectively denote the mean and the standard deviation of bias at the time instance t; μ^{min}_{Bias} and σ^{min}_{Bias} represent that of the minimum bias till time t. We set π as $1.3 \exp(-bias^2) + 0.7$ leading to attain a level of confidence between 68% and 95%. When a new hidden unit/node is added, the attached parameters, i.e. b, V, and W, are randomly sampled from the scope: [-1,1]. The new parameters may also be initialized using adaptive scope selection mechanism [36] to ensure better convergence of the model. To be noted, the hidden unit growing is performed only in the topmost hidden layer at a given time instance.

Hidden unit pruning mechanism. The purpose of hidden unit pruning is to tackle the over-fitting condition of the model, which is attributed by a high variance situation. This can be achieved by reducing the structural complexity of the network, i.e., by decreasing the hidden units in the layers. Thus, the high variance condition for pruning hidden unit can be mathematically formulated in following manner:

$$\mu^t_{Var} + \sigma^t_{Var} \geq \mu^{min}_{Var} + 2\chi \sigma^{min}_{Var} \tag{27}$$

Here μ_{Var}^{t} and σ_{Var}^{t} respectively denote the mean and the standard deviation of the variance at the time instance t; μ_{Var}^{min} and σ_{Var}^{min} represent that of the minimum variance till time t; $\chi = 1.3\exp(-Var) + 0.7$ controls the confidence-level for the sigma rule.

If a high variance condition is identified, the least significant hidden unit in layer l is selected as the candidate to be pruned. The node significance for the i-th hidden node is estimated as the average activation over the elapsed time stamps. Thus, if the m-th unit in l is the pruning candidate, then, $m = \underset{i}{argmin}\left(\lim_{T\to\infty}\sum_{t=1}^{T}\frac{H_l^{i(t)}}{T}\right)$. The overall process of within layer adaptation is summarily presented through the Algorithm 1. The Fig. 3 depicts a typical instance of unfolded SARDINE structure along with hidden unit growing and pruning scenario.

4.3.4 Self-adjustment of network depth. The key objective of layer growing is to learn more complex function for better modeling of the spatial feature evolution at various spatial contexts. Accordingly, whenever there is a drift in spatial context, a new layer is added in the network. The spatial raster time series like satellite remote sensing imagery are usually prone to such drift since these imagery may not be captured at exactly the same temporal condition, and eventually, there remains a high chance of change in spatial context, for example, due to crop harvesting or urban sprawl etc. [39]. Thus, whenever a drift is detected, the addition of new layer can help in better distinguishing between the spatial contexts while modeling the feature evolution.

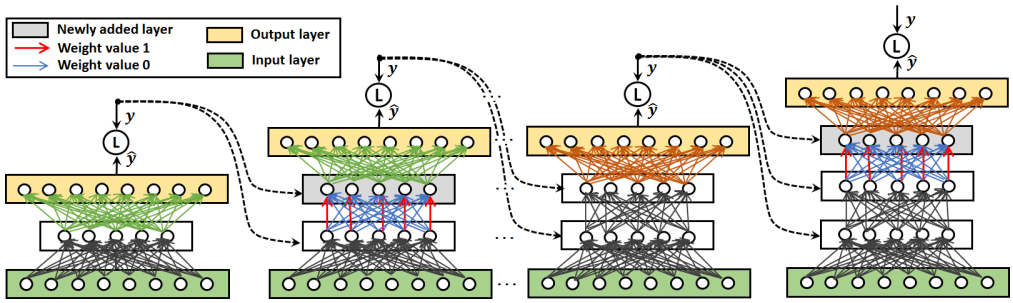


Fig. 5. Self-adjustment of network structure by addition of new hidden layers

Now, in order to detect the drift, we apply the Hoeffding's error bound technique [18], which is primarily proposed for concept drift detection in classification problem. In order to make it fit for the present context of spatio-temporal regression, we modify the accuracy matrix generation process so that it records a value of 1 when the error percentage is less than a desired level (say 2%) and records 0, otherwise. Whenever a drift is detected, the new layer l_{new} is added as per the following rules:

- l_{new} is placed on the top of the current top-most layer l_{curr} and it is initialized with the same number of hidden units as that of l_{curr}, i.e. $e_{l_{new}} = e_{l_{curr}}$;
- the weights between l_{new} and output layer are initialized with the latest weight matrix between l_{curr} and output layer;
- the weights between l_{curr} and l_{new} are initialized with an identity matrix of dimension $[e_{l_{new}} \times e_{l_{new}}]$=$[e_{l_{curr}} \times e_{l_{curr}}]$;

The idea is illustrated through the Fig. 5 and Algorithm 2. To be noted, the creation of a new hidden layer may lead to catastrophic forgetting of previously acquired knowledge

[29]. However, the above-mentioned rules ensure that the previously acquired knowledge is retained during the layer growing process for accommodating the new knowledge. Thus, the proposed SARDINE can be treated as an incremental learning model [1] and effectively used to learn new knowledge from variants of spatial contexts captured through the imagery.

In summary, the proposed SARDINE aims to achieve the desired result with optimal usage of parameters. It does not need to deal with extra parameter like that introduced in terms of gating mechanism in LSTM architecture or in terms of filtering mechanism in CNN.

Algorithm 1: SARDINE_within_layer_updation()

/* This algorithm takes the whole network of SARDINE as input and updates the within layer structure of the topmost hidden layer l */
Estimate $E[\widehat{y}]$ and $E[H]$; /* refer Section 4.3.3 */
Calculate the mean (μ^t_{Bias}) and variance ($(\sigma^t_{Bias})^2$) of network bias; /* see Section 4.3.3 */
Calculate the mean (μ^t_{Var}) and variance ($(\sigma^t_{Var})^2$) of network variance; /* see Section 4.3.3 */
/*—— Growing of Hidden Units in Topmost Layer l ——-*/
if $\left(\mu^t_{Bias} + \sigma^t_{Bias} \geq \mu^{min}_{Bias} + \pi \cdot \sigma^{min}_{Bias}\right)$ **then**
    $e_l \leftarrow (e_l + 1)$;
    $V_l^{new} \leftarrow [-1, 1]$; $W_l^{new} \leftarrow [-1, 1]$; $b_l^{new} \leftarrow [-1, 1]$; /* initiating weights and bias for new unit (H_l^{new}) */
    Reset μ^{min}_{Bias} and σ^{min}_{Bias};
    grow $\leftarrow 1$;
end
else
    $e_l \leftarrow e_l$; grow $\leftarrow 0$;
end
/*—— Pruning of Hidden Units from Topmost Layer l ——-*/
if $\left(\mu^t_{Var} + \sigma^t_{Var} \geq \mu^{min}_{Var} + 2\chi \cdot \sigma^{min}_{Var}\right)$ *&&* $(grow == 0)$ *&&* $(e_l > 1)$ **then**
    Calculate significance of each hidden unit $H^i_l, i \in \{1, 2, \cdots, e_l\}$, /* refer Section 4.3.3 */
    $e_l \leftarrow (e_l - 1)$; /* Eliminate hidden node having the least significance */
    Reset μ^{min}_{Bias} and σ^{min}_{Bias};
    prune $\leftarrow 1$;
end
else
    $e_l \leftarrow e_l$; prune $\leftarrow 0$;
end
if *(grow==1) or (prune==1)* **then**
    Execute forward propagation computation as described in Section 4.3.2;
end
return

Algorithm 2: SARDINE_layer_grow()

/* This algorithm takes the whole network of SARDINE as input and generates an updated network structure with an added hidden layer at the top */
l : Total number of layers in SARDINE;
l_{curr} : Current topmost hidden layer;
$e_{l_{curr}}$: No. of hidden units in the current topmost layer;
$V^{upper} \leftarrow V_{l_{curr}}$;
$l \leftarrow l + 1$;
$l_{new} \leftarrow$ Create_new_layer(); /* contains same number of elements as l_{curr}; i.e. $e_{new} = e_{curr}$*/
$V_{l_{new}} \leftarrow V^{upper}$;
$V_{l_{new}-1} \leftarrow I_{[e_{l_{curr}}]}$; $V_{l_{curr}} \leftarrow V_{l_{new}-1}$; /* $I_{[e_{l_{curr}}]}$ is an identity matrix of dimension $e_{l_{curr}} \times e_{l_{curr}}$ */
return

Algorithm 3: SARDINE-based-Prediction(I, T_{pred})

Input : $I = \{I_1, I_2, \cdots, I_T\}$: Series of observed imagery each containing N pixels denoted by P
Input : T_{pred} = Prediction timestamp = $(T + 1)$
Output : $\widehat{I}$= Predicted image containing predicted pixels $\widehat{P}$

for *all pixels* **do**
  $S(P) \leftarrow spatial\_feature\_representation(I)$; (refer Section 4.2)
end
for *all pixels* **do**
  SARDINE_forward(S[1:T-1]); /* refer SARDINE forward propagation in Section 4.3.2 and consider $t = 1 \cdots T - 1$ */
  if *drift is detected at the end of SARDINE forward propagation* **then**
    SARDINE_layer_grow(); /* refer Section 4.3.4 and Algorithm 2 */
    SARDINE_forward(S[1:T-1]);
  end
  SARDINE_within_layer_adaptation(); /* refer Section 4.3.3 and Algorithm 1 */
  SARDINE_backward(S[1:T-1]); /* refer SARDINE backward propagation in Section 4.3.2 and consider $t = 1 \cdots T - 1$ */
end
for *all pixels* **do**
  $\widehat{S} \leftarrow$ SARDINE_test(S[2:T]); /* primarily this is SARDINE forward propagation using data from $t = 2 \cdots T$ */
  $\widehat{P} \leftarrow$ SIM($\widehat{S}$); /* refer Section 4.4.2; $\widehat{S}$ is predicted spatial feature; $\widehat{P}$ is predicted pixel */
end
return $\widehat{I}$ containing all the predicted pixels $\widehat{P}$

4.4 Module-III: Prediction based on Spatial Feature Influence

While the Module-II of the prediction model learns the temporal evolution of the spatial features, the Module-III executes in parallel to acquire the knowledge of how the spatial features from neighboring locations influence on a central target pixel. Subsequently, the Module-III utilizes this knowledge to finally predict each pixel of the prediction imagery.

4.4.1 Modeling of Spatial Feature Influence. The objective here is to learn a function SIM that can utilize the spatial feature information from the neighboring pixels so as to determine the condition of the target pixel. Mathematically, this can be represented as follows.

$$SIM : \mathbb{R}^M \rightarrow \mathbb{R} \tag{28}$$

where M is the number of neighboring pixels within the spatial neighborhood coverage S such that—

$$P(p, q, t) = SIM(S(P(p, q, t))) \tag{29}$$

where, $P(p, q, t)$ is any pixel condition at time t and $S(P(p, q, t))$ indicates the neighboring spatial features of the pixel.

We learn SIM simply by employing a multilayer perceptron model (MLP) (refer to Fig. 4) with the layer configuration as follows: $[InputLayer : (2d + 1)^2 - 1,\ HiddenLayer : ((2d + 1)^2 - 1)/2, OutputLayer : 1]$. Since we aim to define SIM as a generic function applicable for the entire study zone (captured through the remote sensing image), we use a fixed-layered NN architecture for this purpose, rather than dynamically evolving it.

4.4.2 Prediction. Let the predicted spatial features as generated for any target pixel $P(p, q, t+1)$ from the Module-II is $\widehat{y}$, and the spatial feature influence function parallelly learnt by the Module-III is SIM. Then, predicted value of the pixel becomes:

$$\widehat{P}(p, q, t + 1) = SIM(\widehat{y}) \tag{30}$$

Since the $\widehat{y} = \widehat{S}(p, q, t + 1)$ is generated with consideration to the temporal change in spatial features and the function SIM is defined so as to take into account the influence from neighboring spatial features, on combining the two, the Eq. 30 reflects a spatio-temporal prediction for the pixels in the prediction imagery.

The overall process of proposed SARDINE-based prediction is presented through the Algorithm 3.

5 MODEL EVALUATION

In this section, we validate the proposed SARDINE-based prediction model with respect to a case study on predicting remote sensing data derived from satellite imagery. The details of the dataset, baselines, performance metrics, and the prediction outcomes are thoroughly discussed in the subsequent subsections.

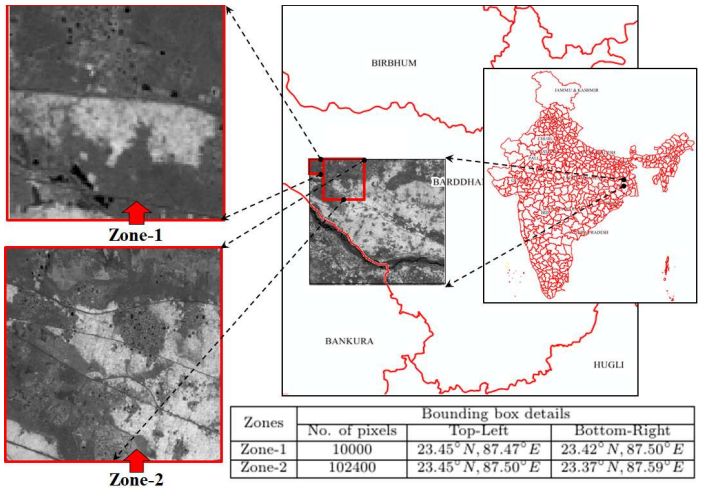
5.1 Dataset: Normalized Difference Vegetation Index (NDVI)

The effectiveness of our proposed SARDINE is evaluated using two sets of *normalized difference vegetation index* or NDVI[1] time series data corresponding to the spatial regions of India and Brazil, respectively, as described below.

- **NDVI Time Series-1:** This consists of annual NDVI imagery covering the Western part of the state of West Bengal, India, during 2004-2011. The dataset is derived from the series of Landsat TM-5 satellite remote sensing imagery having spatial resolution of 30 meter and captured during the Spring season. The same base remote sensing imagery for this dataset is used in [10][13] considering different spatial zone. For the present purpose, the empirical study is carried out considering two randomly selected zones, as indicated by Zone-1 and Zone-2 in Fig. 6 which belong to the district of Bardhaman in the state of West Bengal, India. The primary source of these datasets is the Land Process Distributed Active Archive Center (LPDAAC) of the United States

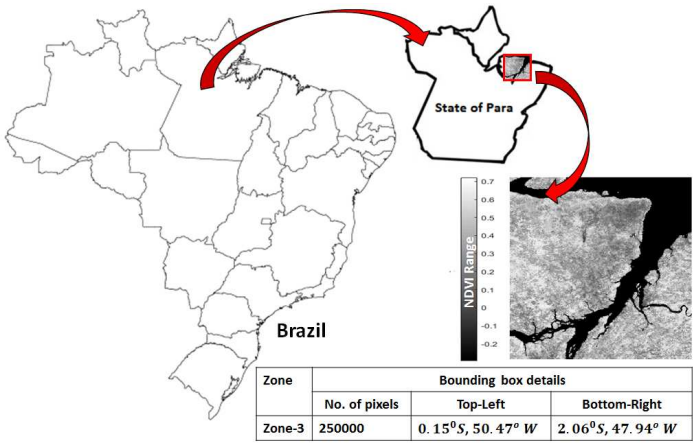
[1] https://earthobservatory.nasa.gov/features/MeasuringVegetation/measuring_vegetation_2.php

Geological Survey (USGS) [35]. Afterwards, we have used ERDAS IMAGINE tool [2] for generating the series of NDVI images from these raw input satellite imagery. Typically, the index is calculated based on the information of band-3 or VIS (visible red radiation) and band-4 or NIR (near-infrared radiation) as follows: $NDVI = \frac{NIR-VIS}{NIR+VIS}$. Training is performed with NDVI data of 2004-2010, and prediction is made for the year 2011.



Zones	Bounding box details		
	No. of pixels	Top-Left	Bottom-Right
Zone-1	10000	$23.45^{\circ}N, 87.47^{\circ}E$	$23.42^{\circ}N, 87.50^{\circ}E$
Zone-2	102400	$23.45^{\circ}N, 87.50^{\circ}E$	$23.37^{\circ}N, 87.59^{\circ}E$

Fig. 6. Study region for NDVI time series-1: District of Bardhaman, West Bengal, India [2]



Zone	Bounding box details		
	No. of pixels	Top-Left	Bottom-Right
Zone-3	250000	$0.15^{0}S, 50.47^{o}W$	$2.06^{0}S, 47.94^{o}W$

Fig. 7. Study region for NDVI time series-2: The State of Para, Brazil, South America

- **NDVI Time Series-2:** This dataset consists of annual NDVI imagery covering the Northern part of Brazil, South America, during 2012-2019. The dataset is derived from the MODIS Terra satellite remote sensing imagery having spatial resolution of 500 meter and captured in the month of January in every considered year. For the

[2]http://www.hexagongeospatial.com/products/remote-sensing/erdas-imagine/overview

present purpose, the empirical study is carried out considering a randomly selected zone, as indicated by Zone-3 in Fig. 7, which belongs to the State of Para, Brazil. The primary source of this dataset is the Land Process Distributed Active Archive Center (LPDAAC) of USGS [35]. The same base NDVI imagery is used in [34] for different spatial zone.

The value of NDVI ranges between -1 and +1, where a zero or negative value of NDVI indicates no vegetation and a value equals to or close to 1 indicates the highest possible density of green leaves. Both the Landsat TM-5 and the MODIS Terra satellite use to capture images on every 16 days. However, in order to avoid the usual seasonal effect and to predict only the change due to deforestation/urbanization in the respective spatial regions, we have considered annual data for both the above-mentioned time series.

5.2 Baseline Methods

The proposed prediction model is evaluated in comparison with eight major baselines, as summarized below.

- **ST-OK** [8]: ST-OK is the spatio-temporal extension of ordinary Kriging [8]. It is widely used for spatio-temporal prediction of missing values in satellite remote sensing imagery;
- **NARNET** [45]: This is a kind of nonlinear auto-regressive neural network model which takes into account spatial features in the form of exogenous variables. NARNET is well-used for time series prediction;
- **MLP** [28]: In our experimentation we have considered multi-layer perceptron (MLP) model as the representative of shallow neural network model. This comparison is necessary to show the effectiveness of using the deep architecture.
- **DSN** [15]: It is known as deep stacking network model. DSN is built on the idea of stacking, where simple modules of functions are developed first and then these are piled over each other to learn more complex functions. It is highly potential for modeling spatial change in time series data.
- **Deep-STEP** [10]: It is a recently proposed deep learning model which is found to show promising performance in spatio-temporal prediction of remote sensing data. The working principle of Deep-STEP model is influenced from vanilla DSN model.
- **LSTM-GP** [44]: LSTM-GP is a state-of-the-art variant of RNN Long-Short-Term-Memory (LSTM) model. Primarily, the model uses a Gaussian Process (GP) layer on the top of LSTM architecture to explicitly account for spatio-temporal dependencies across various data points.
- **CNN-GP** [44]: This is a variant of Convolutional Neural Network (CNN), recently proposed by You et al. [44]. Similar to the previously mentioned LSTM-GP, the CNN-GP model also employs a Gaussian Process (GP) layer, on the top of CNN architecture, to explicitly model the spatio-temporal structure from the input remote sensing data.
- **CNN-LSTM**[3]: In our experimentation, we have also considered a combination of CNN and LSTM models as the baseline to compare with the proposed SARDINE. Our considered CNN-LSTM model is designed with the help of Conv2D, LSTM, and Fully-connected/Dense layers from Keras[2].

All the considered models (except ST-OK) are executed in MATLAB environment in Windows (64-bit OS, 3.10 GHz Intel(R) Xeon(R) CPU processor and 4 GB RAM) using the same set of training and test dataset. For ST-OK, we have used the in-built model

[3]https://keras.io/

Table 1. Generic architectural configuration of the models used in experimentation

Approach	Input unit count	Hidden layer count	Per hidden layer unit count	Output unit count
NARNET	1	2	10	1
MLP	$(2d+1)^2$	2	$(2d+1)^2$	1
DSN	$(2d+1)^2 \times ms$	T (per module one)	$(2d+1)^2$	Topmost module:1 Other modules:$(2d+1)^2$
Deep -STEP	Module-1: $(2d+1)^2$ Others: $(2d+1)^2 \times 2$	T (per module one)	$(2d+1)^2$	Topmost module:1 Other modules:$(2d+1)^2$
LSTM-GP	$(2d+1)^2 - 1$	LSTM: 1 Fully-Connected (FC): 1	LSTM:$\{(2d+1)^2 - 1\} \times 2$ FC:$\{(2d+1)^2 - 1\} \times 2$	LSTM:$(2d+1)^2 - 1$ FC:1
CNN-GP	$(2d+1)^2 - 1$	Convolutional: 3 Fully-Connected (FC): 1	Conv: $\{(2d+1)^2 - 1\} \times 2^{hl}$ FC:$\{(2d+1)^2 - 1\} \times 2$	Conv: $\{(2d+1)^2 - 1\} \times 2^{hl-1}$ FC:1
CNN-LSTM	$(2d+1)^2 - 1$	Convolutional: 2 LSTM: 1 Fully-Connected (FC): 1	Conv:$\{(2d+1)^2 - 1\} \times 2^{hl}$ LSTM:$\{(2d+1)^2 - 1\} \times 2$ FC:$\{(2d+1)^2 - 1\} \times 2$	Conv:$(2d+1)^2 - 1$ LSTM:$(2d+1)^2 - 1$ FC:1
Proposed SARDINE	$(2d+1)^2 - 1$	Automatically decided; Starts from single	Automatically decided; Starts from single	$(2d+1)^2 - 1$ Prediction module:1
d = Degree of spatial neighborhood coverage; T =No. of training images; ms = stacking module sequence number hl = Sequence number of the hidden layer				

in ArcGIS [17]. Regarding the LSTM-GP and CNN-GP models, though originally these were applied on image histograms [44] (instead of being executed on the actual imagery), in our experimentation, we have employed these models directly over our input NDVI imagery, since our objective is to predict the full-image over the same variable, without losing any pixel-information. To be noted, we initially experimented considering both the whole image and the local neighboring extent as the input to the CNN-GP, LSTM-GP, and CNN-LSTM models. However, due to the issue of spatial autocorrelation [43], the models are found to perform better in the latter case and we have considered the same to record their best performance over our datasets. This not only satisfies our problem definition (refer Section 3), but also helps in fair comparison with other competitor models, including our proposed SARDINE. Nevertheless, for each of these models, we have maintained the similar configuration of the base architecture (proportion of convolutional, fully-connected, and LSTM layers), as mentioned in the main paper [44]. In case the performance of the baselines are surprisingly poor, the count of hidden layer units, number of epochs, and other hyper-parameters for these models have been empirically adjusted in multiple executions, to record their best performances. The generic architectural configuration for each of the neural network-based models used in the experimentation is summarized in the Table 1.

5.3 Performance Metrics

The performances of the considered models are evaluated with respect to four evaluation criteria, namely *normalized root mean square deviation* (NRMSD), *mean absolute error* (MAE), *peak signal-to-noise ratio* (PSNR) [16], and *mean structural similarity* (MSSIM) [38] index. The mathematical formulation for each of these metrics is given below.

$$NRMSD = \frac{1}{(O_{max} - O_{min})} \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\mathcal{V}_{o_i} - \mathcal{V}_{p_i})^2} \tag{31}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |\mathcal{V}_{o_i} - \mathcal{V}_{p_i}| \tag{32}$$

$$PSNR = 20.\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{33}$$

$$MSSIM(A, \widehat{A}) = \frac{1}{w}\sum_{i=1}^{w} SSIM(a_i, \widehat{a}_i) \tag{34}$$

In each case, O_{max} is the maximum observed (actual) value of the prediction variable (e.g. NDVI in the present case study), O_{min} is the minimum observed value of the prediction variable, $\mathcal{V}_{o_i}$ is the actual value corresponding to the i-th pixel position, $\mathcal{V}_{p_i}$ is the predicted value corresponding to the i-th pixel position, N is the total number of pixels in the image, MAX_I is the maximum possible pixel value in the image, $\widehat{A}$ and A are the predicted image and original image, respectively, SSIM is the *structural similarity* index [38], $\widehat{a}_i$ and a_i are the contents of the predicted image and original image, respectively, at the i-th local window, and w is the number of local windows considered. The best-fit between actual and predicted image under ideal conditions yields NRMSD = MAE= 0. On the other side, PSNR and MSSIM are used to assess the quality of predicted imagery. The higher the values of PSNR and MSSIM, the better the quality of the predicted image compared to the original one.
Additionally, we have also compared the execution time of proposed SARDINE-based model with that of other considered NN variants, in order to evaluate the utility of online self-adjustment property of SARDINE.

5.4 Results and Discussions

The results of experimental evaluation are summarized in Tables 2-4 and in Figs. 8-13. On analyzing the tables and the figures we can infer the overall effectiveness of the proposed SARDINE-based prediction model as thoroughly explained in the subsequent part of this section.

I. Comparative study with respect to prediction error: The Table 2 shows the comparative study of prediction error in terms of NRMSD and MAE. It is evident from the table that the proposed SARDINE-based prediction outperforms the other baselines in almost all the cases. Though the improvements seem apparently little in terms of absolute values, to be noted, these improvements are quite significant with respect to the considered dataset on NDVI (normalized different vegetation index), the value of which lies in between -1 and +1. Moreover, as discussed in the subsequent part of this section, the comparable performance of

Table 2. Comparative performance study on prediction errors: NRMSD and MAE (boldface indicates the minimum value)

Approach	Study Zones					
	Zone-1		Zone-2		Zone-3	
	NRMSD	MAE	NRMSD	MAE	NRMSD	MAE
ST-OK	0.0365	0.0439	0.0510	0.0522	0.1497	0.1625
NARNET	0.0556	0.0501	0.0655	0.0501	0.1385	0.1462
MLP	0.0334	0.0492	0.0408	0.0474	0.1192	0.1328
DSN	0.0315	0.0426	0.0323	0.0463	0.1016	0.1323
Deep-STEP	0.0296	0.0420	0.0317	0.0451	0.0977	0.1321
LSTM-GP	**0.0262**	0.0428	0.0286	0.0419	0.0728	0.1059
CNN-GP	0.0275	0.0413	0.0301	0.0452	0.1229	0.1408
CNN-LSTM	0.0279	0.0424	0.0297	0.0425	0.0729	0.1106
Proposed (SARDINE)	0.0263	**0.0400**	**0.0273**	**0.0401**	**0.0707**	**0.0914**

Table 3. Comparative performance study on image quality measures: PSNR and MSSIM (boldface indicates the max. value)

Approach	Study Zones					
	Zone-1		Zone-2		Zone-3	
	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM
ST-OK	28.753	0.7962	25.847	0.7751	16.496	0.5811
NARNET	25.105	0.6110	23.671	0.5206	17.169	0.6040
MLP	29.517	0.7242	27.792	0.6476	18.358	0.6498
DSN	30.044	0.8175	29.803	0.7591	19.862	0.7086
Deep-STEP	30.574	0.8379	29.979	0.7645	20.202	0.7117
LSTM-GP	**31.611**	0.8699	30.864	0.8409	22.757	0.8014
CNN-GP	31.215	0.8653	30.150	0.8360	18.209	0.6415
CNN-LSTM	31.092	0.8560	30.213	0.8377	22.745	0.8010
Proposed (SARDINE)	31.601	**0.8701**	**31.277**	**0.8494**	**23.012**	**0.8107**

the state-of-the-art deep learning models (especially LSTM-GP, CNN-GP, and CNN-LSTM) are achieved with notably high computational time and by using empirically adjusted configuration of the respective network models. Further, all these baselines follow the usual convention of multi-pass parameter learning, while the proposed SARDINE-based model achieves the similar performance by scanning the data only once (single-pass data scanning mode). This exhibits the effectiveness of on-the-fly structural adaptation as employed by SARDINE for better adjustment to the spatial/temporal contexts of the various pixels in the imagery. As indicated by the Fig. 8, the percentage improvement of SARDINE is at

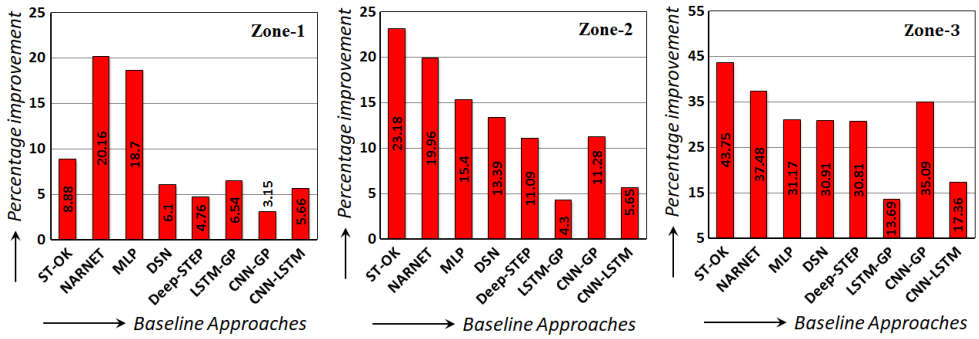


Fig. 8. Percentage improvement of the proposed SARDINE-based prediction model over the absolute error of other baselines

least ∼3%, ∼4%, and ∼14% in case of the Zone-1, Zone-2, and Zone-3, respectively. The improvement statistics reveal that the proposed SARDINE has better capability of handling spatio-temporal diversities in larger zones, compared to the others. The normalized error surfaces, produced by the various models under experimental study, are depicted in Figs. 9-11, from which it is evident that the predictions using SARDINE lead to least amount of error distributions throughout the study zones. Further, though the CNN-GP model is found to perform well in case of smaller area, its performance deteriorates notably with the increasing spatial extents of the zones. This is so, because in order to train the large number of parameters in the CNN models it is necessary to have considerably large number

of input imagery in the sequence. However, in this case study, though the spatial extent of the images over the study zones increases, the count of images per time series remains the same. Accordingly, the CNN models fail to appropriately capture the temporal change in the dataset though the spatial extent of the imagery is increased. For all the considered zones, the overall performance of the LSTM-GP model is found to be better than that of CNN-GP, since the LSTM model can better model the temporal change in the data, and also, with the increasing spatial extent of the imagery, the LSTM model gets an opportunity to improve the learning of spatial dependency. The performance of the CNN-LSTM model, that can exploit both convolutional and LSTM layers, seems quite consistent in every case, though it still cannot outperform the self-adaptive incremental learning of SARDINE.

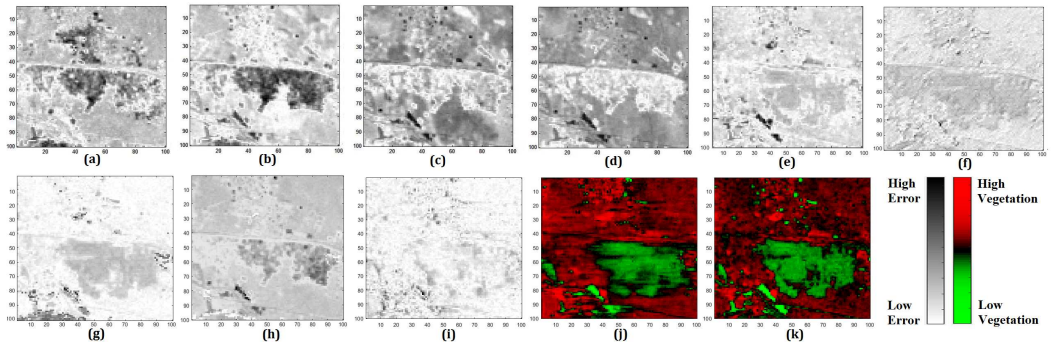


Fig. 9. Comparative study of normalized error surface for the study Zone-1 generated by various approaches: (a) ST-OK, (b) NARNET, (c) MLP, (d) DSN, (e) Deep-STEP, (f) LSTM-GP, (g) CNN-GP, (h) CNN-LSTM, (i) Proposed (SARDINE) (j) Image predicted by SARDINE-based model, (k) Actual image

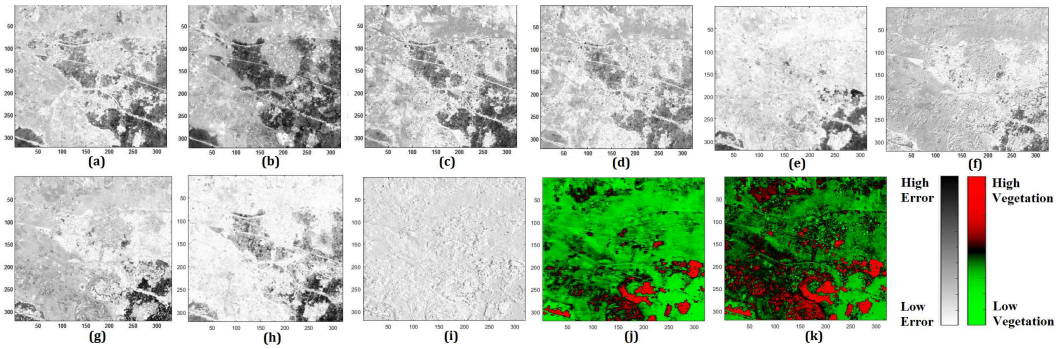


Fig. 10. Comparative study of normalized error surface for the study Zone-2 generated by various approaches: (a) ST-OK, (b) NARNET, (c) MLP, (d) DSN, (e) Deep-STEP, (f) LSTM-GP, (g) CNN-GP, (h) CNN-LSTM, (i) Proposed (SARDINE) (j) Image predicted by SARDINE-based model, (k) Actual image

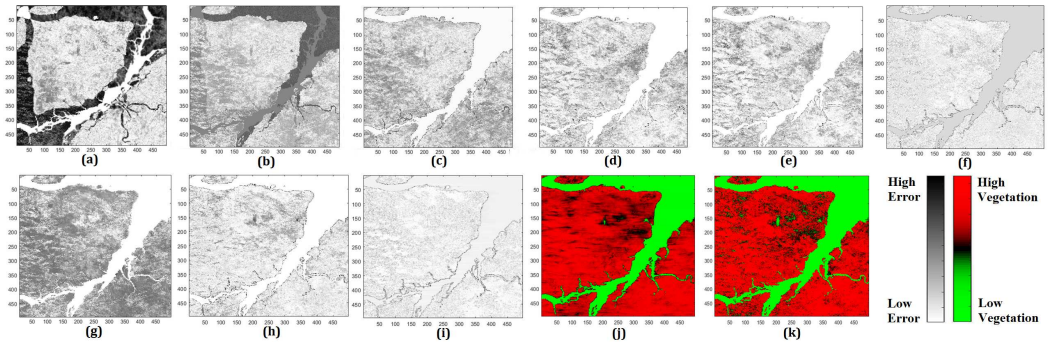


Fig. 11. Comparative study of normalized error surface for the study Zone-3 generated by various approaches: (a) ST-OK, (b) NARNET, (c) MLP, (d) DSN, (e) Deep-STEP, (f) LSTM-GP, (g) CNN-GP, (h) CNN-LSTM, (i) Proposed (SARDINE) (j) Image predicted by SARDINE-based model, (k) Actual image

II. Comparative study with respect to predicted image quality: As mentioned earlier in this section, we have estimated the predicted image quality in terms of PSNR and MSSIM metrics (refer to Table 3). Since the NDVI values range between -1 and +1, we have normalized the estimated PSNR considering the overall range size equating to (1-(-1))=2. The high value of PSNR for our proposed model indicates that the corruption in the predicted pixel values are comparatively less compared to the maximum pixel intensity (NDVI value) in the original image, whereas the high value of MSSIM for the proposed model in all the cases indicates that relative spatial orientation of the pixel-intensities in our predicted imagery remains fairly similar as that of the actual imagery. Overall, it is evident from the Table 3 that the quality of the images predicted using proposed SARDINE is quite acceptable and also these outperform the quality of the images predicted by the baseline techniques, especially ST-OK, NARNET, MLP, DSN, and Deep-STEP. Though the predicted image qualities corresponding to the LSTM-GP, CNN-GP, and CNN-LSTM are sometime similar to that of SARDINE, these state-of-the-art models require larger number of parameter and considerably higher computation time (refer to next section) to attain this performance.

Table 4. Comparative study of execution time (in sec.) of NN-based models [For the baselines (NARNET, MLP, DSN, Deep-STEP, LSTM-GP, CNN-GP, and CNN-LSTM) the time for empirically determining best performing structure is not considered. For the SARDINE, the structure adjustment time is included by default in the execution time]

Image Pixel Count	NARNET	MLP	DSN	Deep-STEP	LSTM-GP	CNN-GP	CNN-LSTM	Proposed Approach without Teacher -forcing	**Proposed (SARDINE)**
10000	43.1×10^2	3.60	74.30	74.24	766.98	280.72	402.63	164.40	128.06
102400	432×10^2	18.64	677.08	667.05	10124.73	6470.90	6825.46	1597.68	1013.34
250000	1330×10^2	154.16	990.87	977.54	18044.52	34432.59	20236.91	2519.57	1885.101

III. Comparative study with respect to computation cost: The results of comparative study on prediction time is shown in the Table 4. Apparently it seems that the computation time required for proposed SARDINE-based prediction is comparatively higher than that of

many of the state-of-the-art prediction models, like MLP, DSN, Deep-STEP etc. However, it must be noted that SARDINE is designed to use only *one time computation* for the predictive analysis. The structural specification is adjusted automatically, as per the data characteristics. No additional empirical execution is required to find the best fit structure of the network. Contrarily, the execution time recorded for the other techniques in Table 4 assumes that the best fit network structure is pre-determined. As a consequence, if we consider all the earlier executions to determine the best structure of these models (especially, DSN, Deep-STEP, LSTM-GP, CNN-GP, and CNN-LSTM), definitely the total execution time will exceed that of the SARDINE. Incidentally, while the Table 4 records the average execution time of the baselines considering a single-run/execution, the prediction results presented through Tables 2-3 and Figs. 8-11 are the best among *minimum* thirty times execution of these baselines (including MLP), considering different hyper-parameters and structural settings. We have not considered multiple-time execution in the Table 4, because the additional time for such empirical study is not fixed and it may vary a lot as per the expertise of the model user. Nevertheless, (even in a single execution) the larger execution time of the LSTM and CNN models are not only led by their multi-pass data-scanning strategy, but also it is due to their large parameter requirement. We also noticed that the computation time increases substantially, when the Gaussian Process (GP) layer is added at the top of LSTM and convolutional layer. In the case of our proposed SARDINE, not only the single-pass data scanning and the on-the-fly structural adaptation properties, but also the use of teacher forcing policy helps to further reduce the *computation time* through increased parallelization, in contrast to the traditional RNN back-propagation through time approach. Besides, the single-pass learning capability ensures that the SARDINE does not need *iteration* over image data/pixels, and accordingly, this drastically reduces the *storage requirement*, since it is not necessary for SARDINE to store the entire image at a time.

IV. Results on Auto-adjustment of network architecture in SARDINE:. As thoroughly described in Section 4, the proposed SARDINE is a self-adaptive deep recurrent model that does not require any predefined architectural specification. The Figs. 12-13 pictorially illustrates this concept with respect to our case study on predicting NDVI imagery for the Zone-1 (Fig. 12(a,d), Fig. 13(a)), Zone-2 (Fig. 12(b,e), Fig. 13(b)), and Zone-3 (Fig. 12(c,f), Fig. 13(c)). It can be noted from Fig. 12(d)- Fig. 12(f) and from Fig. 13(a)- Fig. 13(c) that for every dataset (corresponding to the three study zones), the proposed SARDINE starts with a single hidden layer containing a single hidden unit. Then, it automatically adjusts its layer structure based on the network significance and the change in spatial context of the pixels, calculated from the running error measures (refer Fig. 12(a)- Fig. 12(c)). For the current case study, the proposed SARDINE ends up with 3 hidden layers in case of Zone-1 (see Fig. 12 (d)), 7 hidden layers in case of Zone-2 (see Fig. 12(e)), and 5 hidden layers in case of Zone-3 (see Fig. 12(f)). In each case, the layer size (number of hidden units per layer) is dynamically and optimally set with a value of 4 (in case of Zone-1 and Zone-2) or 6 (in case of Zone-3) through the node growing and pruning mechanism (refer to Fig. 13(a)- Fig. 13(c)). The dynamic structural adaptability of SARDINE improves the overall *scalability* of the proposed prediction model.

Overall, though the prediction quality of the proposed SARDINE is quite comparable with that of the state-of-the-art models, the SARDINE has added benefits with respect to high flexibility and low computational cost, especially compared to the existing deep learning models. The self-adjustment property and incremental learning power further helps SARDINE to become scalable over diverse datasets.

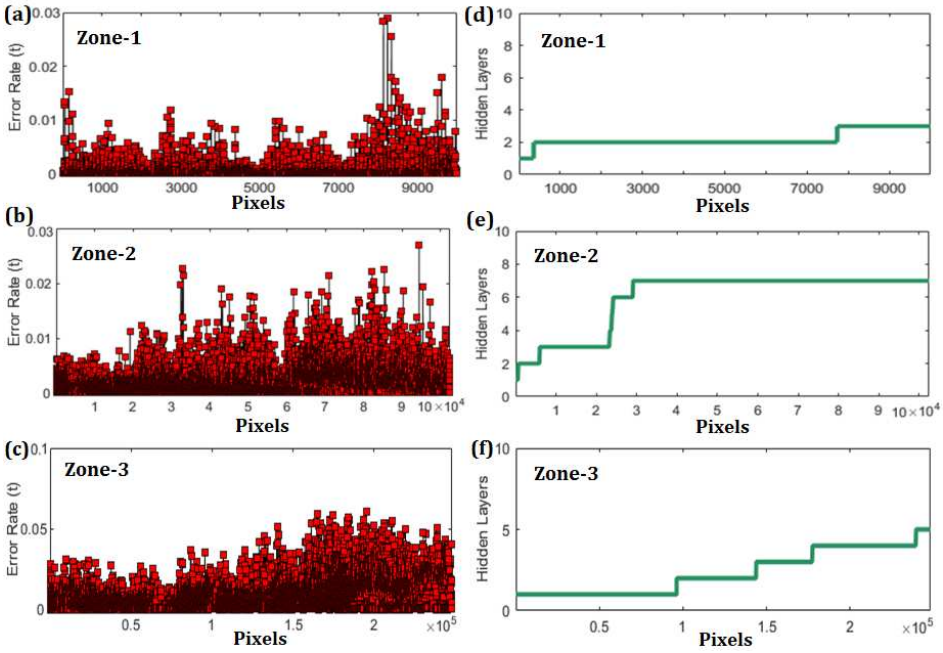


Fig. 12. Adjustment of hidden-layer count ((d)-(f)) according to the drift defined in terms of error ((a)-(c))

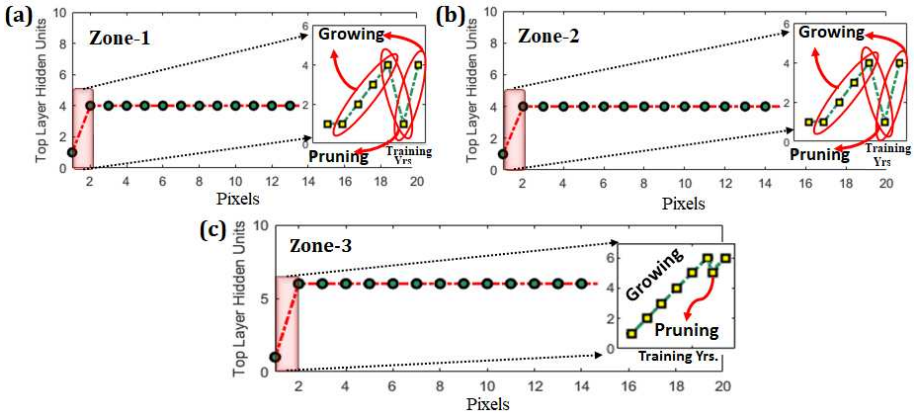


Fig. 13. Typical scenario of within layer architecture adjustment for SARDINE

6 CONCLUSION

In this work we have proposed SARDINE, a variant of deep recurrent neural network model for spatio-temporal prediction of derived remote sensing imagery. The novelties in this work are threefold. First, the proposed SARDINE does not require empirical analysis to determine the best network configuration at a particular prediction scenario and is able to automatically adapt its network architecture as per the characteristics of the input imagery. Second, the

SARDINE learns online and in single-pass manner, which makes it appropriate to deal with derived remote sensing imagery containing millions of pixels. Third, the proposed SARDINE is able to learn new knowledge on spatial evolution, without relinquishing the previously acquired knowledge, by using optimal set of parameters and without employing complex gating mechanism. Further, the use of teacher forcing policy helps SARDINE to achieve improved parallelization during training process and subsequently reduces the computational time requirement. Due to the intrinsic property of single-pass learning, self-evolution, and optimal parameter usage, SARDINE is found to have huge potentials to be applied for the purpose of nowcasting from remotely sensed data. The incremental learning ability of SARDINE also makes it suitable for dealing with large spatial imagery, since the model can retain old knowledge for applying in similar spatial contexts without attempting to re-learn from the scratch. Incidentally, the proposed SARDINE-based prediction model is not only applicable for the satellite remote sensing imagery but is generic and potential enough to be applied on any other kind of image streams. Ample scope remains in further improving the proposed model with embedded online feature extraction mechanism that can be achieved through added convolutional layers in the architecture.

REFERENCES

[1] Andri Ashfahani and Mahardhika Pratama. 2019. Autonomous Deep Learning: Continual Learning Approach for Dynamic Environments. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 666–674.

[2] Bhuvan. 2016. *Indian Geo-Platform of ISRO, India*. http://bhuvan.nrsc.gov.in. [Online; December 2016].

[3] Jayajit Chakraborty. 2011. Revisiting Toblers first law of geography: Spatial regression models for assessing environmental justice and health risk disparities. In *Geospatial analysis of environmental health*. Springer, 337–356.

[4] Bin Chen, Bo Huang, Lifan Chen, and Bing Xu. 2017. Spatially and temporally weighted regression: a novel method to produce continuous cloud-free Landsat imagery. *IEEE Transactions on Geoscience and Remote Sensing* 55, 1 (2017), 27–37.

[5] Qing Cheng, Huanfeng Shen, Liangpei Zhang, Qiangqiang Yuan, and Chao Zeng. 2014. Cloud removal for remotely sensed images by similar pixel replacement guided with a spatio-temporal MRF model. *ISPRS journal of photogrammetry and remote sensing* 92 (2014), 54–68.

[6] Hannah M Cooper, Qi Chen, Charles H Fletcher, and Matthew M Barbee. 2013. Assessing vulnerability due to sea-level rise in Maui, Hawai i using LiDAR remote sensing and GIS. *Climatic Change* 116, 3-4 (2013), 547–563.

[7] José Luis Crespo, Marta Zorrilla, Pilar Bernardos, and Eduardo Mora. 2007. A new image prediction model based on spatio-temporal techniques. *The Visual Computer* 23, 6 (2007), 419–431.

[8] Noel Cressie and Christopher K Wikle. 2015. *Statistics for spatio-temporal data*. John Wiley & Sons.

[9] Monidipa Das and Soumya K Ghosh. 2016. A cost-efficient approach for measuring Moran's index of spatial autocorrelation in geostationary satellite data. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 5913–5916.

[10] Monidipa Das and Soumya K Ghosh. 2016. Deep-STEP: A Deep Learning Approach for Spatiotemporal Prediction of Remote Sensing Data. *IEEE Geoscience and Remote Sensing Letters* 13, 12 (2016), 1984–1988.

[11] Monidipa Das and Soumya K Ghosh. 2017. A Deep-Learning-Based Forecasting Ensemble to Predict Missing Data for Remote Sensing Analysis. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10, 12 (2017), 5228–5236.

[12] Monidipa Das and Soumya K Ghosh. 2017. Measuring Moran's I in a Cost-Efficient Manner to Describe a Land-Cover Change Pattern in Large-Scale Remote Sensing Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10, 6 (2017), 2631–2639.

[13] Monidipa Das and Soumya K Ghosh. 2019. Space-time Prediction of High Resolution Raster Data: An Approach based on Spatio-temporal Bayesian Network (STBN). In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*. ACM, 129–135.

[14] Monidipa Das, Mahardhika Pratama, Andri Ashfahani, and Subhrajit Samanta. 2019. FERNN: A fast and evolving recurrent neural network model for streaming data classication. In *2019 International Joint Conference on Neural Networks (IJCNN)*. 1–8.
[15] Li Deng and Dong Yu. 2014. Deep learning: Methods and applications. *Foundations and Trends in Signal Processing* 7, 3–4 (2014), 197–387.
[16] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2016. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 2 (2016), 295–307.
[17] ESRI. 2017. ArcGIS for Desktop. http://www.esri.com/software/arcgis/arcgis-for-desktop. [Online; Accessed 12-Dec-2017].
[18] Isvani Frías-Blanco, José del Campo-Ávila, Gonzalo Ramos-Jiménez, Rafael Morales-Bueno, Agustín Ortiz-Díaz, and Yailé Caballero-Mota. 2015. Online and non-parametric drift detection methods based on Hoeffdings bounds. *IEEE Transactions on Knowledge and Data Engineering* 27, 3 (2015), 810–823.
[19] Florian Gerber, Rogier de Jong, Michael E Schaepman, Gabriela Schaepman-Strub, and Reinhard Furrer. 2018. Predicting Missing Values in Spatio-Temporal Remote Sensing Data. *IEEE Transactions on Geoscience and Remote Sensing* 56, 5 (2018), 2841–2853.
[20] Rafael C Gonzalez, Richard E Woods, et al. 2002. Digital image processing [M]. *Publishing house of electronics industry* 141, 7 (2002).
[21] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.
[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[23] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. 2016. Combining satellite imagery and machine learning to predict poverty. *Science* 353, 6301 (2016), 790–794.
[24] CY Ji. 2008. Haze reduction from the visible bands of LANDSAT TM and ETM+ images over a shallow water reef environment. *Remote Sensing of Environment* 112, 4 (2008), 1773–1783.
[25] Jintao Ke, Hai Yang, Hongyu Zheng, Xiqun Chen, Yitian Jia, Pinghua Gong, and Jieping Ye. 2018. Hexagon-Based Convolutional Neural Network for Supply-Demand Forecasting of Ride-Sourcing Services. *IEEE Transactions on Intelligent Transportation Systems* (2018).
[26] Chao-Hung Lin, Po-Hung Tsai, Kang-Hua Lai, and Jyun-Yuan Chen. 2013. Cloud removal from multitemporal satellite images using information cloning. *IEEE transactions on geoscience and remote sensing* 51, 1 (2013), 232–241.
[27] Frank Silvio Marzano, Giancarlo Rivolta, Erika Coppola, Barbara Tomassetti, and Marco Verdecchia. 2007. Rainfall nowcasting from multisatellite passive-sensor images using a recurrent neural network. *IEEE Transactions on Geoscience and Remote Sensing* 45, 11 (2007), 3800–3812.
[28] Binh Thai Pham, Dieu Tien Bui, Hamid Reza Pourghasemi, Prakash Indra, and MB Dholakia. 2017. Landslide susceptibility assesssment in the Uttarakhand area (India) using GIS: a comparison study of prediction capability of naïve bayes, multilayer perceptron neural networks, and functional trees methods. *Theoretical and Applied Climatology* 128, 1-2 (2017), 255–273.
[29] Mahardhika Pratama, Choiru Za'in, Andri Ashfahani, Yew Soon Ong, and Weiping Ding. 2019. Automatic construction of multi-layer perceptron network from streaming examples. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1171–1180.
[30] Atiqur Rahman, Shiv Prashad Aggarwal, Maik Netzband, and Shahab Fazal. 2011. Monitoring urban sprawl using remote sensing and GIS techniques of a fast growing urban centre, India. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 4, 1 (2011), 56–64.
[31] Fernando Sedano, Pieter Kempeneers, and George Hurtt. 2014. A Kalman filter-based method to generate continuous time series of medium-resolution NDVI images. *Remote Sensing* 6, 12 (2014), 12381–12408.
[32] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. 2017. Deep learning for precipitation nowcasting: A benchmark and a new model. In *Advances in Neural Information Processing Systems*. 5617–5627.
[33] K Sowmya, CM John, and NK Shrivasthava. 2015. Urban flood vulnerability zoning of Cochin City, southwest coast of India, using remote sensing and GIS. *Natural Hazards* 75, 2 (2015), 1271–1286.
[34] Vandana Tomar, Vinay Prasad Mandal, Pragati Srivastava, Shashikanta Patairiya, Kartar Singh, Natesan Ravisankar, Natraj Subash, and Pavan Kumar. 2014. Rice equivalent crop yield assessment using MODIS sensors based MOD13A1-NDVI data. *IEEE Sensors Journal* 14, 10 (2014), 3599–3605.

[35] USGS-EarthExplorer. 2019. Land Processes Distributed Active Archive Center. https://lpdaac.usgs.gov/data_access/usgs_earthexplorer. [Online; August 2019].

[36] Dianhui Wang and Ming Li. 2017. Stochastic configuration networks: Fundamentals and algorithms. *IEEE transactions on cybernetics* 47, 10 (2017), 3466–3479.

[37] Laigang Wang, Yongchao Tian, Xia Yao, Yan Zhu, and Weixing Cao. 2014. Predicting grain yield and protein content in wheat by fusing multi-sensor and multi-temporal remote-sensing images. *Field Crops Research* 164 (2014), 178–188.

[38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

[39] Geoffrey I Webb, Loong Kuan Lee, Bart Goethals, and François Petitjean. 2018. Analyzing concept drift and shift from sample data. *Data Mining and Knowledge Discovery* 32, 5 (2018), 1179–1199.

[40] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*. 802–810.

[41] Yuting Yang, Junyu Dong, Xin Sun, Estanislau Lima, Quanquan Mu, and Xinhua Wang. 2017. A CFCC-LSTM model for sea surface temperature prediction. *IEEE Geoscience and Remote Sensing Letters* 15, 2 (2017), 207–211.

[42] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. 2019. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *AAAI Conference on Artificial Intelligence*.

[43] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[44] Jiaxuan You, Xiaocheng Li, Melvin Low, David Lobell, and Stefano Ermon. 2017. Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data.. In *AAAI*. 4559–4566.

[45] Lijing Yu, Lingling Zhou, Li Tan, Hongbo Jiang, Ying Wang, Sheng Wei, and Shaofa Nie. 2014. Application of a new hybrid model with seasonal auto-regressive integrated moving average (ARIMA) and nonlinear auto-regressive neural network (NARNN) in forecasting incidence cases of HFMD in Shenzhen, China. *PloS one* 9, 6 (2014), e98241.

[46] Qin Zhang, Hui Wang, Junyu Dong, Guoqiang Zhong, and Xin Sun. 2017. Prediction of sea surface temperature using long short-term memory. *IEEE Geoscience and Remote Sensing Letters* 14, 10 (2017), 1745–1749.