Predict Land Covers with Transition Modeling and Incremental Learning

Xiaowei Jia[1], Ankush Khandelwal[1], Guruprasad Nayak[1], James Gerber[2],
Kimberly Carlson[3], Paul West[2], Vipin Kumar[1]

[1]Department of Computer Science and Engineering, University of Minnesota

[2]Institute on the Environment, University of Minnesota

[3]Department of Natural Resources and Environmental Management, University of Hawai'i Mānoa

[1]jiaxx221@umn.edu, {ankush,nayak,kumar}@cs.umn.edu, [2]{jsgerber,pcwest}@umn.edu,
[3]kimberly.carlson@hawaii.edu

Abstract

Successful land cover prediction can provide promising insights in the applications where manual labeling is extremely difficult. However, traditional machine learning models are plagued by temporal variation and noisy features when directly applied to land cover prediction. Moreover, these models cannot take fully advantage of the spatio-temporal relationship involved in land cover transitions. In this paper, we propose a novel spatio-temporal framework to discover the transitions among land covers and at the same time conduct classification at each time step. Based on the proposed model, we incrementally update the model parameters in the prediction process, thus to mitigate the impact of the temporal variation. Our experiments in two challenging land cover applications demonstrate the superiority of the proposed method over multiple baselines. In addition, we show the efficacy of spatio-temporal transition modeling and incremental learning through extensive analysis.

1 Introduction

The need for discovering land use and land cover (LULC) changes is ever growing to understand the global environment and climate change. In this paper we investigate the land cover prediction problem, which aims to learn the land cover patterns using available ground-truth from history (e.g. before 2010) and then automatically predict land covers in more recent years (e.g. after 2010) when ground-truth is not available.

Many existing land cover products still heavily rely on visual interpretation of satellite imagery - which takes advantage of human expertise in the land cover detection process [14,17]. However, these products have several limitations for long-term monitoring. First, the human resources needed for large-scale annual digitizing are substantial. Moreover, the visual interpretation may result in both false positives and false negatives due to the quality of images and the observational mistakes. Therefore the products heavily reliant on visual interpretation are usually not available in recent years or not in good quality.

Major recent advances in access to and processing of remote sensing data provide opportunities for a long-term land cover monitoring over large regions. With the frequently available remotely sensed multi-spectral data over the entire globe (e.g. MODIS and Landsat), it becomes possible to use machine learning methods to automatically predict land covers in recent years.

An intuitive solution for land cover prediction is to train a traditional classifier in previous years using available ground-truth and then directly apply the learned classifier in the following years. However, land cover prediction is commonly plagued by severe data heterogeneity and noisy spectral features [11]. A large region usually contains a variety of land covers, and some land covers, e.g. forest and plantation, are highly confused with each other. Moreover, there exists strong temporal variation over the years. As a result, even the same land cover shows different multi-spectral features in different years. Furthermore, the traditional models are severely affected by the noise in spectral features, which is especially common in tropical areas. Due to these reasons, the model learned from previous years would perform poorly in the following years.

To overcome these challenges, we propose a predictive spatio-temporal model to mitigate these impacts from two aspects. First, we model the transition of land covers in our classification process. The transition modeling from both temporal and spatial perspectives assists not only in capturing the temporal variation, but also in alleviating the noise in spectral features by making temporally and spatially consistent classification. The transition modeling is further combined with the state-of-the-art deep learning methods to extract discriminative knowledge to distinguish different land covers. Second, we introduce an incremental learning strategy that updates model parameters in the prediction process. More specifically, we utilize a sliding window to incorporate latest spectral features and adapt the model to the temporal variation. An EM-style algorithm is then proposed to update the model parameters as we move the sliding window.

We extensively evaluate our method on two real-world applications - detecting plantations in Indonesia and identifying burned areas in Montana state, US. The results confirm that the proposed method outperforms multiple baselines in predicting land covers. Also, we provide illustrative examples and case studies to demonstrate the efficacy of our spatio-temporal framework and the incremental learning strategy.

2 Problem Definition

In this problem, we are provided with a set of locations/pixels $X = \{x_1, x_2, ..., x_N\}$ as well as their multi-spectral features at $T + m$ time steps (i.e. years), $x_i = \{x_i^1, x_i^2, ..., x_i^T, x_i^{T+1}, ..., x_i^{T+m}\}$, for $i = 1$ to N (detailed feature description given in Section 4). Besides, to include the spatial information, we represent the neighborhood of i^{th} location as $N(i)$. It is noteworthy that the neighborhood can be adjusted in different applications. In this work we define $N(i)$ to be the set of locations within a 1500 meters by 1500 meters squared range centered at i^{th} location. In addition, we have the ground-truth labels for each location i until the time step T, as $y_i = \{y_i^1, ..., y_i^T\}$ for $i = 1$ to N.

Our objective is to learn a predictive classification model using the available ground-truth from the time step 1 to T, and subsequently apply the model to estimate the labels from $T + 1$ to $T + m$.

2.1 Challenges In land cover prediction, traditional classification models are crippled due to the temporal variation. Consider a location i with a fixed land cover type (i.e. y_i^t stays the same over time), it can still have different spectral features x_i^t in different years due to varying precipitation, sunlight, and temperature. Therefore the patterns learned before T may not conform to the land cover patterns at following time steps. Consequently, the classification performance at $T + j$ will decrease drastically as j increases from 1 to m.

Moreover, the classification process can be greatly impacted by the noisy input features. The spectral features are noisy due to multiple reasons, including natural disturbances (e.g. cloud, fog, smoke, etc.) and data acquisition errors.

3 Method

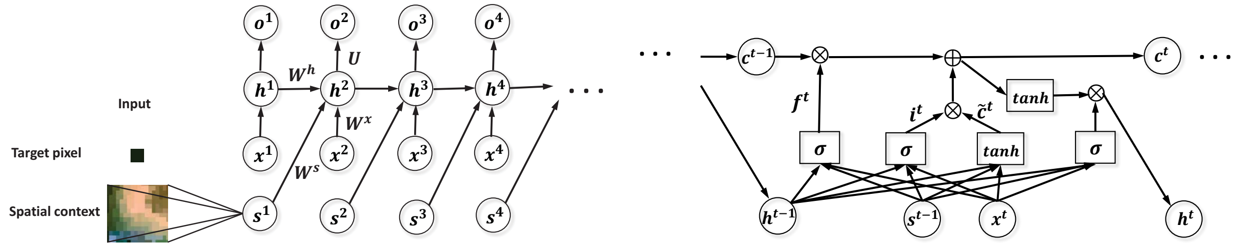
In this section, we first introduce our proposed spatio-temporal framework. Then based on the proposed framework, we discuss the extraction of spatial context features in Section 3.2 and the use of the proposed method in land cover prediction in Section 3.3.

3.1 Spatio-temporal modeling Rather than focusing on a single location or time step, we propose to combine both spatial and temporal information in a unified framework. The spatio-temporal information can assist in discovering land covers in the following aspects:
1. Different land covers have different temporal transition patterns. For instance, forests can be logged and then get converted to croplands/urban area while the opposite transitions from croplands/urban area to forest rarely happen.
2. The spatial context can provide useful insights into the expansion/shrinkage of a specific land cover. For instance, the new plantations/croplands are usually cultivated around the existing plantations/croplands. Some land covers caused by external factors, e.g. burned areas caused by forest fire, are also likely to propagate to surrounding locations.
3. Due to the noisy spectral features, the classification separately conducted on each individual location at each time step frequently leads to classification errors. By combining the spatio-temporal information, we can mitigate the noise and make the classification outputs consistent over space and over time.

Recent advances in deep learning models enable automatic extraction of discriminative knowledge from multi-spectral features thus to better distinguish different land covers. In this work we propose the variants of two deep learning models - Recurrent Neural Network (RNN) and Long Short-term Memory (LSTM) to incorporate both spatial and temporal information. Since LSTM is in effect a special and extended version of RNN, we first introduce our proposed framework based on RNN, which is depicted in Fig. 1 (a). Given the noisy spectral features in complex high-dimensional feature space, we wish to learn a more discriminative representation to recognize our desired land cover classes. Therefore we introduce the latent representation h^t, which is mapped from the input features through a weight matrix W^x. Besides, we include the temporal and spatial dependence to generate the hidden representation h^t at time step t. More formally, we generate h^t using the following equation. Hereinafter we omit the subscript index i when it causes no ambiguity.

$$h^t = tanh(W^h h^{t-1} + W^x x^t + W^s s^{t-1}), \tag{3.1}$$

where W^h, W^x and W^s represent the weight matrices that connect h^t to h^{t-1}, x^t, and s^{t-1} respectively, and s^{t-1} denotes the spatial context features at time $t - 1$. We will defer the discussion of spatial context features



(a) (b)

Figure 1: The structure of our proposed spatio-temporal method: (a) The flow chart of proposed spatio-temporal model based on RNN, and (b) the variant of LSTM cell in our proposed method.

in Section 3.2. Here the bias terms are omitted as they can be absorbed into the weight matrix.

We can observe that both h^{t-1} and s^{t-1} are utilized to generate h^t. On one hand, W^h aims to capture the transition from a temporal perspective. For instance, "forest→cropland" is more likely to occur than "forest→water". On the other hand, the weight matrix W^s is utilized to learn the relation between the spatial context at $t-1$ and the land cover at t. A target location is more likely to convert to certain land covers (e.g. plantation, cropland and burned area) if there exist such land covers in the neighborhood at $t-1$. More importantly, the spatial dynamics of land covers also depend on the properties of surrounding locations, e.g., burned area is more likely to propagate along the direction of high greenness level. Therefore we extract spatial context features from the neighborhood locations at $t-1$ to guide the spatial dynamics.

Based on h^t, we can classify the land covers at each time step t using a softmax function:

$$p(o^t) = softmax(Uh^t), \tag{3.2}$$

where U is the weight matrix between h^t and the classification output o^t. The model can be easily extended to deep structure [21] and the parameters can be estimated by back-propagation algorithm.

While RNN-based method can model the land cover transition and classify the land cover types, it gradually loses the connection to long history as time progresses [1]. Therefore RNN-based method may fail to grasp the transition pattern for some long-term cyclical events such as El Niño. To this end we also propose a variant of LSTM to classify land cover types. LSTM is better in modeling long-term dependencies where each time step needs more contextual information from the past. The difference between our proposed LSTM and RNN-based models only lies in the generation of hidden representation h^t. In LSTM-based model, the input features and h^t are connected via an LSTM cell, as shown in Fig. 1 (b).

Here we briefly introduce the LSTM cell in our proposed method. Each LSTM cell contains a cell state c^t, which serves as a memory and forces the hidden variables h^t to reserve information from the past. To generate the cell state c^t, we combine the cell state c^{t-1} at $t-1$ and the information at current time step. The transition of cell state over time forms a memory flow, which enables the modeling of long-term dependencies.

Specifically, we first generate a new candidate cell state $\tilde{c}^t$ by combining the input features x^t, the previous hidden layer h^{t-1} and the spatial context features s^{t-1} into a $tanh(\cdot)$ function, as follows:

$$\tilde{c}^t = tanh(W_h^c h^{t-1} + W_x^c x^t + W_s^c s^{t-1}), \tag{3.3}$$

where W_h^c, W_x^c, and W_s^c denote the weight parameters used to generate candidate cell state. Then a forget gate layer and an input gate layer are generated as follows:

$$\begin{aligned} f^t &= \sigma(W_h^f h^{t-1} + W_x^f x^t + W_s^f s^{t-1}), \\ i^t &= \sigma(W_h^i h^{t-1} + W_x^i x^t + W_s^i s^{t-1}), \end{aligned} \tag{3.4}$$

where $\{W_h^f, W_x^f, W_s^f\}$ and $\{W_h^i, W_x^i, W_s^i\}$ denote two sets of weight parameters for generating forget gate layer f^t and input gate layer i^t, respectively. $\sigma(\cdot)$ denotes the sigmoid function, and therefore each entry in the forget/input gate layer ranges in [0,1]. The forget gate layer is used to filter the information inherited from c^{t-1}, and the input gate layer is used to filter the candidate cell state at time t. In this way we obtain the new cell state c^t as follows:

$$c^t = f^t \otimes c^{t-1} + i^t \otimes \tilde{c}^t, \tag{3.5}$$

where $\otimes$ denotes entry-wise product.

Finally, the hidden representation is generated using a hidden gate layer e^t and the obtained cell state, as:

$$\begin{aligned} e^t &= \sigma(W_h^e h^{t-1} + W_x^e x^t + W_s^e s^{t-1}), \\ h^t &= e^t \otimes tanh(c^t), \end{aligned} \tag{3.6}$$

where W_h^e, W_x^e and W_s^e are the weight parameters that are used to generate hidden gate layer, which determines the information to output from c^t to h^t.

3.2 Spatial context feature extraction In our proposed spatio-temporal framework, the spatial context features assists in guiding the transition among different land covers and also in mitigating the noise in spectral features. While there exist a variety of unsupervised feature extraction methods that aim to learn a representative feature space [10, 16], the generated embeddings from these methods do not contain enough discriminative knowledge. Since the spatial context features are associated with land cover transitions, the extraction process should be more related with the supervised information of land cover transitions. For instance, the spatial context features of "forest→forest" and "forest→plantation" should be different.

Here the input for i^{th} location is the raw spectral features of the locations in the neighborhood $N(i)$ at time t, and we wish to learn a nonlinear mapping with parameter γ to the spatial context features s_i^t. We first define a probabilistic model similar with Neighborhood Component Analysis (NCA) [4]. For each location i, it connects to another location j with a probability as:

$$p_{ij} = \frac{exp(-d_{ij}^2)}{\sum_{j'} exp(-d_{ij'}^2)}, \tag{3.7}$$

where d_{ij} is the Euclidean distance between s_i^t and s_j^t, as $d_{ij} = ||s_i^t - s_j^t||$.

On the other hand, we define a target distribution using land cover labels, as follows:

$$q_{ij} = \frac{exp(-l_{ij}^2)}{\sum_{j'} exp(-l_{ij'}^2)}. \tag{3.8}$$

The target distance function l_{ij} is defined using the supervised label information, as follows:

$$l_{ij} = \begin{cases} \infty, & y_i^t \neq y_j^t \,\&\, y_i^{t+1} \neq y_j^{t+1}, \\ max(\delta_i, \delta_j)\sqrt{(d_y^t)^2 + (d_y^{t+1})^2}, & otherwise, \end{cases} \tag{3.9}$$

where the label y^t is in one-hot representation and $d_y^t = ||y_i^t - y_j^t||$. The target distance is only defined between two transitions with at least one common land cover at either t or $t+1$. $\delta_i = p(y_i^{t+1}|y_i^t)$, measuring the fraction of locations with label y_i^t at time t to be converted to y_i^{t+1} at $t+1$. Since popular transitions are usually more interesting, we utilize δ_i and δ_j to measure the popularity of the land cover transition. According to the equation, the popular transitions have a larger target distance with other transitions. Now we wish to have the distribution defined by s^t to be close to the distribution defined by the target distance.

Formally, we aim to minimize the Kullback-Leibler (KL) divergence between two distributions p and q. Our objective is expressed as follows:

$$\min_{\gamma} KL = \sum_{i,j} q_{ij} log \frac{q_{ij}}{p_{ij}}. \tag{3.10}$$

The gradient can then be computed as:

$$\begin{aligned} \frac{\partial KL}{\partial \gamma} &= \sum_i \frac{\partial KL}{\partial s_i^t} \frac{\partial s_i^t}{\partial \gamma} \\ &\propto -\sum_i \frac{\partial s_i^t}{\partial \gamma} \sum_j (s_i^t - s_j^t)(l_{ij} - p_{ij}). \end{aligned} \tag{3.11}$$

The derivative $\frac{\partial s_i^t}{\partial \gamma}$ can be estimated by back-propagation if we adopt a neural network structure to generate spatial context features. The computation of p and q can be time-consuming given large data size. Therefore in our implementation we cluster the data in each transition type and randomly sample a set of locations from each cluster. Then we will extract spatial context features using the sampled locations.

3.3 Incremental Learning in Prediction In this section, we will introduce the incremental learning strategy in prediction process. For simplicity we discuss the prediction process using the RNN-based model, which, however, can be easily extended to LSTM-based model by replacing the generation of h^t. Given the learned model, an intuitive way to predict the label from $T+1$ to $T+m$ is to recursively follow Eqs. 3.1 and 3.2. However, due to the temporal variation, many land covers frequently change their spectral features over the years. Since the learned model does not incorporate the variation of spectral features from $T+1$ to $T+m$, it may perform poorly when directly applied at these time steps. Therefore the model cannot be used by itself to predict the future land covers.

To conquer the temporal variation, we propose an incremental learning strategy. Basically, we wish that the model can be updated with the latest spectral features and become aware of the land cover transition in more recent years. The proposed incremental learning method can be summarized in an Expectation-Maximization (EM) process. In E-step we first generate the predicted label $\hat{y}^{t+1}$ at next time step $t+1$, where $t = T$ to $T+m-1$. Here we use $\hat{y}$ to distinguish the predicted labels from the provided labels y. Specifically, we sample the predicted label following a multinomial distribution $P(\hat{y}^{t+1}|x^{1:t+1};\theta)$, where θ denotes the model parameters learned from 1 to t, as follows:

$$\begin{aligned} &h^{t+1} = tanh(W^h h^t + W^x x^{t+1} + W^s s^t), \\ &\hat{y}^{t+1} \sim Mult(softmax(Uh^{t+1})). \end{aligned} \tag{3.12}$$

Then in M-step, we utilize a sliding temporal window with length w to capture the temporal variation. The sliding window aims to include the time steps that are representative for next prediction. We hold an assumption that the temporal variation between consecutive time steps is much smaller than the temporal variation against long history. Hence before we conduct prediction at $t+2$, we first move the sliding window to cover the time steps from $t-w+2$ to $t+1$. Then we update the model parameters using the samples in the sliding window. Here the label of $t+1$ is obtained from the prediction process in E-step. An example of update procedure from $T+1$ to $T+2$ is shown in Fig. 2. More formally, in M-step we update parameters as follows:

$$\theta^{new} = argmin_{\theta} L([y^{t-w+2:T}, \hat{y}^{T+1:t+1}], o^{t-w+2:t+1}), \tag{3.13}$$

where $[\cdot,\cdot]$ denotes the concatenation of two sets of labels, $L(\cdot)$ denotes the cross-entropy loss function for softmax output. Basically, we wish to minimize the difference between the classification outputs $o^{t-w+2:t+1}$ and the obtained labels in the sliding window. We can observe that when $t-w+2$ is greater than T, we rely simply on the predicted labels without using provided ground-truth. The whole incremental learning process is depicted in Algorithm 1. The time complexity of the prediction process is $O(lmN)$, where l is a constant determined by the dimension of input features, spatial context features and hidden representation.

Algorithm 1 Incremental learning in prediction.

```
Input: ${x^1, x^2, ..., x^T, x^{T+1}, ..., x^{T+M}}$: A series of input features.
    ${y^1, ..., y^T}$: A series of label.
    The mapping to generate spatial features.
    The learned model from $t = 1$ to $T$.
Output: ${\hat{y}^{T+1:T+m}}$
for time step $t \leftarrow T$ to $T+m-1$ do
    Generate spatial context features $s^t$.
    Estimate $\hat{y}^{t+1}$ by Eq. 3.12.
    Move sliding window, include $\hat{y}^{t+1}$ as training labels.
    Update model parameters by Eq. 3.13 using $y^{t-w+2:T}$, $\hat{y}^{T+1:t+1}$, $x^{t-w+2:t+1}$ and $s^{t-w+1:t}$.
end for
```

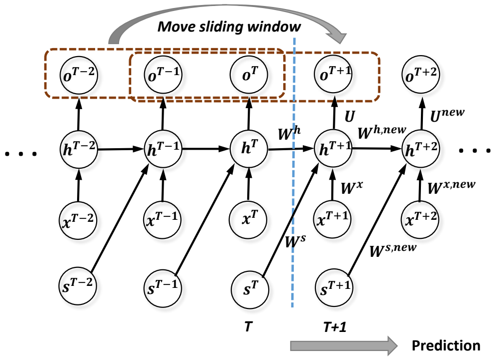


Figure 2: The incremental learning process from T to $T+1$ - a sliding window of length 3 is moved to include the predicted labels in $T+1$. Then the updated parameters are utilized to predict $T+2$.

While the incremental learning strategy incorporates the spectral features from recent years, it results in much computational cost before making prediction. Therefore in real applications with large data, we wish to avoid frequently updating parameters. In this work we consider the possibility to stop the incremental learning at an early stage.

Specifically, we adopt the convergence rate (CR) metric in sequential labeling [15] to determine the necessity of the incremental training at a particular time step t. To estimate convergence rate, we keep a small validation set and manually label the samples in it. The convergence rate is then computed as $\sum_{i\in val,t=T+1:T+m} \eta_{i,t}/(mn_v)$. Here n_v denotes the number of locations in the validation set val, and m represents the number of time steps under prediction. $\eta_{i,t}$ is the probability for the i^th instance to be correctly classified at t^{th} time step in prediction. We measure the convergence rate after each round of parameter update, and we will stop the incremental learning if the convergence rate is above a threshold. Then we directly apply the learned model to predict labels at the remaining time steps.

4 Experiment

In this section we will evaluate our method on two real applications - plantation monitoring in Indonesia and burned area detection in Montana state, US. Specifically, we utilize the 500-meter resolution MODIS data product, which consists of 7 reflectance bands (620-2155 nm) collected by MODIS instruments onboard Aqua and Terra satellites. This product provides images at a global scale for every 8 days. For each year we take 15 most discriminative composite images according to domain knowledge (e.g. land covers are hardly distinguishable during winter). We concatenate the 7-band spectral features of the selected 15 images as the yearly input for the learning model. Our goal is to predict yearly land cover labels during a specific period.

4.1 Oil palm plantation in Indonesia

The industry of oil palm plantation is a key driver for deforestation in Indonesia. Since oil palm plantations frequently have similar properties (e.g. greenness) with tropical forest, it is difficult to automatically monitor the plantation for long period. In our experiment we utilize two latest datasets - RSPO dataset [5] and Tree Plantation dataset [17] to create ground-truth. RSPO dataset labels each location as one of multiple pre-defined land cover types including oil palm, forest, grassland, etc. It is available on 2000, 2005, and 2009. In contrast, Tree Plantation dataset only labels plantation locations in 2014. We combine both two datasets and utilize Enhanced Vegetation Index (EVI) time series from 2001 to 2014 to manually create yearly ground-truth for 27,923 locations in Kalimantan region of Indonesia through 2001-2014. Each location is labeled as one of the categories from {"plantation", "forest", "other"}.

We utilize 40% of the locations as training locations, and only the ground-truth of 2001 to 2009 is used for training. Another 20% of the locations are kept as validation set for selecting hyper-parameters. The remaining 40% of the locations serves as the test locations. We measure the prediction performance through 2010 to 2014 on both training locations and test locations. The prediction performance is measured using F1-score and Area Under the Curve (AUC) of plantation class. Now we first introduce the methods involved in our tests:

Artificial Neural Networks (ANN): In this baseline we train a global ANN model based on the training data from 2001 to 2009. Then we directly apply the learned model to predict the land covers during 2010-2014.

Recurrent Neural Networks (RNN): We train an RNN model using only the spectral features of training locations. Then we directly apply it for prediction.

RNN+spatial context features (sRNN): We combine the spectral features and the spatial context features into the RNN-based model, as described earlier. Then we use it for prediction.

sRNN+incremental learning (siRNN): We combine sRNN and incremental learning in prediction.

Long short-term memory (LSTM): Here we train an LSTM model using only the spectral features of training locations. Then we apply it for prediction.

Convolutional long short-term memory (cvLSTM): Rather than extracting spatial context features, we directly utilize the spectral features from the neighborhood at $t-1$ as s^{t-1}. This is equivalent to learning a convolutional model on the neighborhood of each location at $t-1$, but replacing the target location with its spectral features at t.

LSTM+spatial context features (sLSTM): Similar with sRNN, we combine the spatial context features and LSTM, and directly use it for prediction.

sLSTM+incremental learning (siLSTM): We combine sLSTM and incremental learning in prediction.

We show the performance of each method on both training and test locations in Table 1. It can be observed that siLSTM model brings around 33% improvement than ANN model, which lies in the ignorance of spatio-temporal information by ANN. The comparison between the methods with and without spatial context features (sRNN vs. RNN, sLSTM vs. LSTM) clearly reveals the advantage brought by the spatial context information. The improvement from cvLSTM to sLSTM also demonstrates the effectiveness of our method in extracting spatial context features. In cvLSTM, the training process is dominated by large amount of spectral features from the neighborhood while ignoring the information from the target location. On the other hand, the use of incremental learning leads to a significant improvement from sRNN/sLSTM to siRNN/siLSTM with $w = 5$. Moreover, by modeling land cover transition, the methods other than ANN result in less difference between training and test locations. Finally the result reveals the superiority of LSTM based model over RNN based model, which mainly stems from the capacity of LSTM in reserving long-term memory and in modeling more complex transition relationship.

It can also be observed that the prediction performance of each method decreases from 2010 to 2014, which is caused by the temporal variation. To better illustrate this, we retrain LSTM-based methods only using the ground-truth until 2005 and predict on 2010-2014. According to Table 2, the performance greatly drops compared to Table 1. However, siLSTM still outperforms other methods by a considerable margin.

Then we evaluate how the capacity of hidden representation affects the performance. In Fig. 3 (a), we measure the performance in F1-score with different number of hidden variables. The large number of hidden variables will lead to overfitting while too few hidden variables lack the capacity to well model the transition.

Another interesting finding from Table 1 is that the reduction of AUC over time is not as much as the reduction of F1-score. Therefore we conclude that as time progresses, most plantation locations still have larger probability to be classified as plantation than the other non-plantation locations. However, the plantation class gradually gets confused with other classes and the absolute probability value decreases. Such phenomenon can be utilized to validate the effectiveness of proposed incremental learning strategy. In particular, we measure the average confidence on the real plantation locations over the years. Here the confidence denotes the probability for our model to classify each location as plantation, which can be obtained from softmax output. We show the average confidence computed using sRNN, siRNN, sLSTM and siLSTM, in Fig. 3 (b). By incorporating latest information to update model in the prediction process, the incremental methods can better characterize the plantation class in recent years, and consequently result in higher confidence.

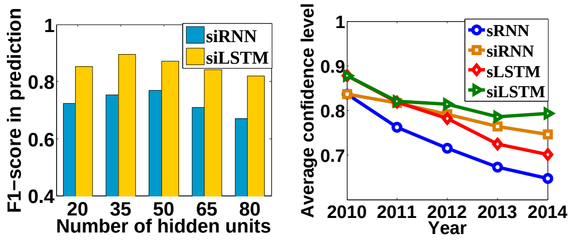
To show the efficacy of incremental learning after each round of parameter update, we measure CR after each year with incremental learning, as shown in Table 1. The CR value is measured through 2010-2014 using the validation set, and the initial value in 2009 does not involve incremental learning. It is clear that CR increases as the model incrementally learn new knowledge from 2010 to 2014, but shows shows limited improve-

Table 1: The plantation prediction performance of each method in 2010-2014, measured in F1-score (AUC). The prediction is conducted on both training locations and test locations.

Method	Set	2010	2011	2012	2013	2014
ANN	train	0.838 (0.767)	0.724 (0.738)	0.714 (0.718)	0.684 (0.695)	0.677 (0.693)
	test	0.772 (0.732)	0.703 (0.706)	0.678 (0.702)	0.647 (0.680)	0.643 (0.677)
RNN	train	0.876 (0.773)	0.807 (0.758)	0.738 (0.741)	0.715 (0.708)	0.703 (0.695)
	test	0.866 (0.770)	0.797 (0.758)	0.736 (0.730)	0.695 (0.701)	0.690 (0.693)
sRNN	train	0.902 (0.787)	0.830 (0.761)	0.769 (0.749)	0.752 (0.746)	0.720 (0.742)
	test	0.898 (0.784)	0.816(0.761)	0.746 (0.727)	0.738 (0.745)	0.719 (0.731)
siRNN	train	0.902 (0.787)	0.852 (0.767)	0.804 (0.763)	0.795 (0.751)	0.768 (0.756)
	test	0.898 (0.784)	0.863 (0.765)	0.797 (0.755)	0.783 (0.750)	0.769 (0.745)
LSTM	train	0.885 (0.775)	0.867 (0.757)	0.817 (0.718)	0.756 (0.704)	0.771 (0.703)
	test	0.876 (0.780)	0.870 (0.753)	0.803 (0.701)	0.761 (0.699)	0.758 (0.692)
cvLSTM	train	0.897 (0.783)	0.880 (0.774)	0.833 (0.762)	0.797 (0.738)	0.778 (0.726)
	test	0.889 (0.781)	0.873 (0.769)	0.844 (0.761)	0.776 (0.729)	0.763 (0.725)
sLSTM	train	0.938 (0.820)	0.922 (0.803)	0.872 (0.782)	0.831 (0.783)	0.793 (0.765)
	test	0.937 (0.802)	0.903 (0.782)	0.867 (0.770)	0.813 (0.762)	0.794 (0.744)
siLSTM	train	0.938 (0.820)	0.937 (0.806)	0.914 (0.798)	0.911 (0.794)	0.898 (0.781)
	test	0.937 (0.802)	0.936 (0.805)	0.919 (0.791)	0.905 (0.792)	0.895 (0.776)

Table 2: The plantation prediction performance of each method in 2010-2014, measured in F1-score (AUC). Only the ground-truth during 2001-2005 is used in this test.

Method	Set	2010	2011	2012	2013	2014
LSTM	train	0.391 (0.741)	0.409 (0.706)	0.380 (0.664)	0.286 (0.639)	0.309 (0.650)
	test	0.385 (0.736)	0.398 (0.695)	0.355 (0.657)	0.288 (0.637)	0.282 (0.631)
cvLSTM	train	0.452 (0.751)	0.412 (0.706)	0.369 (0.674)	0.309 (0.637)	0.355 (0.649)
	test	0.442 (0.740)	0.406 (0.703)	0.369 (0.673)	0.306 (0.639)	0.339 (0.640)
sLSTM	train	0.464 (0.766)	0.415 (0.727)	0.405 (0.662)	0.378 (0.680)	0.390 (0.664)
	test	0.447 (0.754)	0.405 (0.719)	0.366 (0.680)	0.360 (0.661)	0.349 (0.667)
siLSTM	train	0.587 (0.763)	0.541 (0.699)	0.544 (0.700)	0.436 (0.667)	0.423 (0.670)
	test	0.582 (0.761)	0.541 (0.694)	0.534 (0.700)	0.397 (0.677)	0.420 (0.649)



(a) (b)

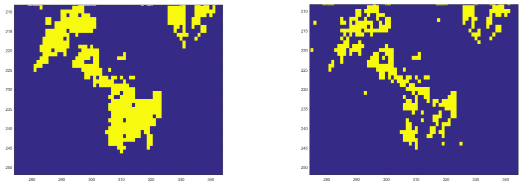
Figure 3: (a) The performance (F1-score) on test locations measured with different number of hidden variables. (b) The change of average confidence on real plantation locations over time.

Table 3: The progression of CR in plantation detection as we incrementally update model over years.

Methods	2009	2010	2011	2012	2013	2014
siRNN	0.778	0.781	0.793	0.804	0.804	0.807
siLSTM	0.786	0.882	0.918	0.930	0.933	0.934

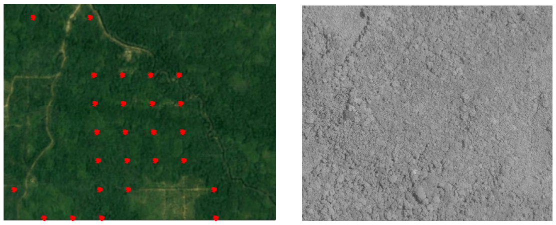
ment after 2012. Hence we can terminate the incremental learning after 2012 to save computational cost.

In Fig. 4, we show the efficacy of incorporating spatio-temporal information by an example region. Compared to ANN, sLSTM generates more compact plantation regions and stays resistant to noisy labels. As shown in Fig. 5, the region is marked as plantation by ground-truth but the proposed method classifies it as forest. The high-resolution image from Digital Globe verifies that this region is indeed forest.



(a) (b)

Figure 4: (a) A region with detected plantations by sLSTM. (b) The same region with ANN detection. The yellow color denotes the detected plantations.



(a) (b)

Figure 5: A region (red) that is correctly labeled as forest by our method, but labeled as plantation by ground-truth. (a) The region in Google earth image. (b) The high-resolution image from Digital Globe.

4.2 Burned area detection in Montana

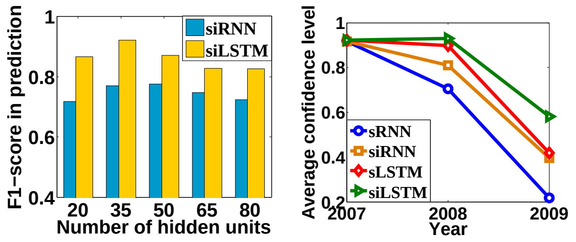
In this part we evaluate our proposed framework in detecting burned area in Montana. we obtained fire validation data until 2009 from government agencies responsible for monitoring and managing forests and wildfires. Each burned location is associated with the time of burning. In total we select 15,107 MODIS locations and each location has a binary label of burned/not burned on every year from 2001 to 2009.

We divide the data in the same proportion with our test in plantation application. Here we train each method using the ground-truth until 2006 and predict on 2007-2009. From the results shown in Table 4, we can observe that the spatio-temporal information and the incremental learning bring considerable improvement. On the other hand, we can find that all the methods have low F1-scores in 2009. Such phenomenon lies in that the burned locations in 2009 are very few, and both precision and recall will be severely disturbed if the model makes any classification error.

Table 4: The prediction of burned area in 2007-2009, measured in F1-score (AUC).

Method	Set	2007	2008	2009
ANN	train	0.844 (0.775)	0.546 (0.842)	0.059 (0.801)
	test	0.786 (0.770)	0.487 (0.830)	0.059 (0.797)
RNN	train	0.901 (0.780)	0.605 (0.955)	0.084 (0.817)
	test	0.866 (0.770)	0.603 (0.969)	0.057 (0.793)
sRNN	train	0.974 (0.790)	0.643 (0.971)	0.104 (0.821)
	test	0.972 (0.780)	0.641 (0.974)	0.096 (0.815)
siRNN	train	0.974 (0.790)	0.853 (0.995)	0.232 (0.881)
	test	0.972 (0.780)	0.776 (0.994)	0.204 (0.865)
LSTM	train	0.907 (0.781)	0.629 (0.967)	0.160 (0.818)
	test	0.908 (0.781)	0.616 (0.971)	0.145 (0.796)
cvLSTM	train	0.912 (0.794)	0.714 (0.970)	0.168 (0.826)
	test	0.909 (0.792)	0.705 (0.970)	0.166 (0.806)
sLSTM	train	0.981 (0.800)	0.820 (0.973)	0.256 (0.841)
	test	0.980 (0.792)	0.804 (0.985)	0.247 (0.816)
siLSTM	train	0.981 (0.800)	0.925 (0.997)	0.357 (0.905)
	test	0.980 (0.792)	0.922 (0.997)	0.325 (0.900)

Similar with plantation test, we measure the performance with different number of hidden variables, as displayed in Fig. 6 (a). We obtain a similar hill-shaped pattern with that in plantation test. Then we validate the incremental learning method by measuring the change of average confidence, as shown in Fig. 6 (b). We can see that the confidence of siRNN and siLSTM decrease slower than siRNN and siLSTM, respectively.



(a) (b)

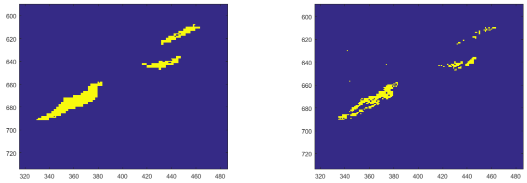
Figure 6: (a) The performance (F1-score) on test locations measured with different number of hidden variables. (b) The change of average confidence on real burned locations over time.

Table 5: The progression of CR in burned area detection as we conduct incremental learning over time.

Methods	2006	2007	2008	2009
siRNN	0.578	0.603	0.614	0.631
siLSTM	0.649	0.677	0.683	0.697

Moreover, we show the progression of CR in Table 5. It is clear that every incremental learning step increases CR. Therefore we conclude that the temporal variation ubiquitously exists during 2007-2010 and we should not terminate incremental learning early.

In Fig. 7, we show an example region with detected burned area using ANN and sLSTM. The detected burned locations using sLSTM show a more spatially consistent pattern than the locations detected by ANN, since ANN suffers much from the noisy features and the temporal variation.



(a) (b)

Figure 7: (a) A region with detected burned locations by sLSTM. (b) The same region with ANN detection. The yellow color denotes the detected burned locations.

5 Related Works

Discovering LULC changes is essential for understanding environment and climate change [18]. Recent advances in collecting remote sensing data have spawned many research works on monitoring land covers in large regions [2, 6, 8, 20], e.g., the semi-automated monitoring of global changes in tree canopy cover [6].

However, there are still many challenges in identifying certain land covers. For e.g., even the most widely used global forest product [6] defines forest on the basis of tree structure and therefore does not differentiate between forest and plantations. The main difficulty lies in that tree plantations frequently have spectral properties similar to natural forests [3]. On the other hand, detecting burned area is also challenging since the fires have a seasonal pattern and only last for a few months. In addition, the prediction of these land covers become even more challenging due to the temporal variation [11].

Conventional machine learning models have been widely explored in a variety of land cover prediction problems [7, 22]. However, these method have limited capacity to extract discriminative information and capture the spatio-temporal relationship from large amount of remotely sensed data. The last decade has witnessed the surge of deep learning in a variety of real-world applications [9,12,13,19]. While RNN and LSTM have shown promising performance in sequence labeling, their application in land cover discovery is still limited. For instance, [13] explores the transferable temporal relationship from RNN and applies it on land cover change detection. However, these methods do not make fully use of spatio-temporal information in modeling land cover transitions. Besides, they overlook the contextual information which provides advantage in classification task [23]. More importantly, when used in land cover prediction, these methods are vulnerable to temporal variation and noisy spectral features.

6 Acknowledgement

This work was funded by the NSF Awards 1029711 and 0905581, and the NASA Award NNX12AP37G. Access to computing facilities was provided by NASA Earth Exchange and Minnesota Supercomputing Institute.

7 Conclusions

In this work we propose a spatio-temporal framework for land cover prediction. We first combine the spatial context features and temporal relationship to discover the land cover transitions. Then we utilize the incremental learning to incorporate knowledge from more recent years, and adapt the learned model to the varying spectral features. The experimental results on plantation and burned area detection demonstrate the effectiveness of both the transition modeling and the incremental learning strategy. Besides, the proposed method has potential to contribute to a much larger community of land cover discovery problems and to assist in understanding global environment and climate change.

References

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 1994.

[2] K. M. Carlson, L. M. Curran, G. P. Asner, A. M. Pittman, S. N. Trigg, and J. M. Adeney. Carbon emissions from forest conversion by kalimantan oil palm plantations. *Nature Climate Change*, 2013.

[3] H. Fan, X. Fu, Z. Zhang, and Q. Wu. Phenology-based vegetation index differencing for mapping of rubber plantations using landsat oli data. *Remote Sensing*, 2015.

[4] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. Salakhutdinov. Neighbourhood components analysis. In *NIPS*, 2004.

[5] P. Gunarso, M. E. Hartoyo, F. Agus, Killeen, T. J., and J. Goon. Roundtable on sustainable palm oil, kuala lumpur, malaysia. *Reports from the technical panels of the 2nd greenhouse gas working group of RSPO*, 2013.

[6] M. C. Hansen, P. V. Potapov, R. Moore, M. Hancher, S. Turubanova, A. Tyukavina, D. Thau, S. Stehman, S. Goetz, et al. High-resolution global maps of 21st-century forest cover change. *Science*, 2013.

[7] C. Homer, C. Huang, et al. Development of a 2001 national land-cover database for the united states. *Photogrammetric Engineering & Remote Sensing*, 2004.

[8] X. Jia, A. Khandelwal, J. Gerber, K. Carlson, P. West, and V. Kumar. Learning large-scale plantation mapping from imperfect annotators. In *IEEE BigData*, 2016.

[9] X. Jia, X. Li, K. Li, V. Gopalakrishnan, G. Xun, and A. Zhang. Collaborative restricted boltzmann machine for social event recommendation. In *ASONAM*, 2016.

[10] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[11] A. Karpatne, Z. Jiang, R. R. Vatsavai, S. Shekhar, and V. Kumar. Monitoring land-cover changes: A machine-learning perspective. *IEEE Geoscience and Remote Sensing Magazine*, 2016.

[12] X. Li, X. Jia, H. Li, H. Xiao, J. Gao, and A. Zhang. Drn: Bringing greedy layer-wise training into time dimension. 2015.

[13] H. Lyu, H. Lu, and L. Mou. Learning a transferable change rule from a recurrent neural network for land cover change detection. *Remote Sensing*, 2016.

[14] J. Miettinen, C. Shi, W. J. Tan, and S. C. Liew. 2010 land cover map of insular southeast asia in 250-m spatial resolution. *Remote Sensing Letters*, 2012.

[15] W. Min, B. Mott, J. Rowe, B. Liu, and J. Lester. Player goal recognition in open-world digital games with long short-term memory networks.

[16] M. Norouzi, M. Ranjbar, and G. Mori. Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning. In *CVPR*, 2009.

[17] R. Petersen, E. Goldman, N. Harris, S. Sargent, D. Aksenov, A. Manisha, E. Esipova, V. Shevade, T. Loboda, N. Kuksina, et al. Mapping tree plantations with multispectral imagery: preliminary results for seven tropical countries. *World Resources Institute*, 2016.

[18] R. A. Pielke. Land use and climate change. *Science*, 2005.

[19] T. Sauter, B. Weitzenkamp, and C. Schneider. Spatio-temporal prediction of snow cover in the black forest mountain range using remote sensing and a recurrent neural network. *International Journal of Climatology*, 2010.

[20] A. Shalaby and R. Tateishi. Remote sensing and gis for mapping and monitoring land cover and land-use changes in the northwestern coastal zone of egypt. *Applied Geography*, 2007.

[21] I. Sutskever. *Training recurrent neural networks*. PhD thesis, University of Toronto, 2013.

[22] Q. Wu, H.-q. Li, R.-s. Wang, et al. Monitoring and predicting land use change in beijing using remote sensing and gis. *Landscape and urban planning*, 2006.

[23] G. Xun, X. Jia, V. Gopalakrishnan, and A. Zhang. A survey on context learning. *TKDE*, 2016.