Incremental Dual-memory LSTM in Land Cover Prediction

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

Land cover prediction is essential for monitoring global environmental change. Unfortunately, traditional classification models are plagued by temporal variation and emergence of novel/unseen land cover classes in the prediction process. In this paper, we propose an LSTM-based spatio-temporal learning framework with a dual-memory structure. The dual-memory structure captures both long-term and short-term temporal variation patterns, and is updated incrementally to adapt the model to the ever-changing environment. Moreover, we integrate zero-shot learning to identify unseen classes even without labelled samples. Experiments on both synthetic and real-world datasets demonstrate the superiority of the proposed framework over multiple baselines in land cover prediction.

KEYWORDS

LSTM; land cover; zero-short learning

1 INTRODUCTION

Many governments, companies and non-government organizations (NGOs) are increasingly interested in identifying Land Use and Land Cover (LULC) changes since they are associated with global environmental change and human-environment interactions [5, 21]. The monitoring of LULC changes requires the ability to map land covers over large regions and over a long period.

Many existing land cover mapping products are manually created through visual interpretation, which takes advantage of human expertise in the labeling process [6, 20]. However, the limitations of this approach are manifold. First, manual labeling may result in both false positives and false negatives due to observational mistakes. Second, this approach usually requires multiple researchers to delineate land covers, likely resulting in inconsistency among observers. Most importantly, the required substantial human resources make it infeasible for large regions or for a long period. Therefore, high-quality land cover mapping products created by manual labeling are only available in specific years in history and usually cannot cover recent years due to the time expense of visual interpretation.

In contrast, we focus on automated land cover monitoring and develop a classification model to map land covers in recent years. In this way, we allow the scientific domain researcher to analyze the latest land cover conditions and land cover changes. Specifically, assume we have the manually created ground-truth in history (e.g. before 2010), we aim to train a classification model to learn the land cover patterns from ground-truth data, and then predicts land covers in more recent years (e.g. after 2010) when ground-truth is not available. With the frequently available remotely sensed multi-spectral data over the entire globe, it becomes possible to learn the mapping relation from spectral features to land covers, and then apply this learned relation to predict land covers.

Compared with traditional classification problems, the land cover prediction process is hampered by data heterogeneity [11]. Specifically, the data heterogeneity exists in several aspects. First, the spectral features of land covers are different in different regions. Therefore, a local model is usually trained on each target region [11]. Second, the spectral features of each land cover can change over time. Such temporal variation is mainly caused by changes in temperature, sunlight and precipitation in different years, and potentially leads to misclassification. Furthermore, unseen/novel land cover classes may appear during the prediction process. Considering a target region consisting of forests and croplands with available ground-truth before 2010, if some forest locations are converted to urban areas after 2010, the learned model from local training data until 2010 needs to predict urban class without having access to relevant training data.

To solve these challenges, we propose a novel framework which combines multiple types of features into a classification model. In particular, we extract seasonal features and spatial context features from the multi-spectral data at each location, and then project them to the temporal features of corresponding land cover. The temporal features are extracted from a large set of observed land cover series and constitute a "semantic" temporal feature space. Inspired by zero-shot learning [19, 23], we can relate unseen land cover classes to the temporal feature space and identify them.

KDD'17, August 13–17, 2017, Halifax, NS, Canada.
 ISBN 978-1-4503-4887-4/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3097983.3098112

Then we integrate this learning approach into a long-short term memory (LSTM) model, which aims to predict land covers at each time step, while also learning the transitions among land covers. To better capture the land cover transitions and temporal variation patterns, we propose to extend the conventional LSTM using a dual-memory structure. In particular, we partition the transition and variation knowledge into a long-term memory and a short-term memory. The long-term memory is responsible for preserving the information from long history while the short-term memory can capture the patterns in more recent time steps. During the prediction process, we incrementally update the dual-memory structure via an Expectation-Maximization (EM) process. Also, we refine the temporal features of land cover classes as time progresses. On one hand, such update process can reduce the impact of temporal variation. On the other hand, the refinement on land cover classes can mitigate the projection shift [2] that occurs on unseen classes.

We extensively evaluate the performance of our proposed framework on synthetic and real-world datasets. The validation confirms the effectiveness of our proposed method in tackling temporal variation and in identifying both existing and unseen land cover classes. Moreover, we compare the generated product by our framework to the existing manually created plantation product in Indonesia, and demonstrate that our framework ensures better quality.

Our contributions can be summarized as:

- We propose multiple types of representative features in land cover applications and combine these features in a classification model.
- We integrate zero-shot learning approach into a proposed dual-memory LSTM framework. In this way, the proposed framework is capable of predicting unseen classes while also modeling the spatio-temporal dependencies in both long-term and short-term events.
- An EM-style incremental update strategy is adopted to address the temporal variation.
- We evaluate our proposed method in multiple land cover applications. The results demonstrate that our method can be utilized to generate high-quality land cover mapping products for scientific research.

2 PROBLEM DEFINITION

In this section, we formalize the land cover prediction problem and introduce notation. In particular, we focus on a set of N locations/pixels, indexed by $i = 1$ to N. For each location i, we have its spectral features at $T + m$ time steps (i.e. years), $z_i = \{z_i^1, z_i^2, ..., z_i^T, z_i^{T+1}, ..., z_i^{T+m}\}$, for $i = 1$ to N. Each dimension in spectral features represents the reflectance value at a specific bandwidth. In our method discussion we omit the index i when it causes no ambiguity. To utilize the spatial context information, we represent the neighborhood of location i as $N(i)$. It is noteworthy that the neighborhood can be adjusted for different applications. In addition, we have the ground-truth labels for each location i from $t = 1$ to T, denoted as $l_i = \{l_i^1, ..., l_i^T\}$.

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 $T+m$. Note that due to the temporal variation (discussed in Section 1), the land cover patterns after T may not conform to the learned patterns before T, which significantly degrades the classification performance from $T + 1$ to $T + m$. Moreover, as the spectral features greatly change over space [11], the classification model is expected to be trained locally on the target region. Hence, the available ground-truth in the target region may contain limited number of land covers and may not cover all the land covers that would appear after T. For example, if some forest locations convert to plantations after T while no plantation exists in this region before T, the learned model cannot identify plantations.

Finally, the spectral features contain much noise due to natural disturbances (e.g. cloud, fog, smoke, etc.) and data acquisition errors. The spectral features collected in each year also show seasonal cyclic changes.

3 METHOD

In this section we start with the description of the proposed zero-shot learning model and the involved feature representation. Then we integrate this model into a dual-memory LSTM framework, i.e., we replace the projection in LSTM with the projection defined in zero-shot learning model. We then introduce the framework and discuss the incremental update strategy.

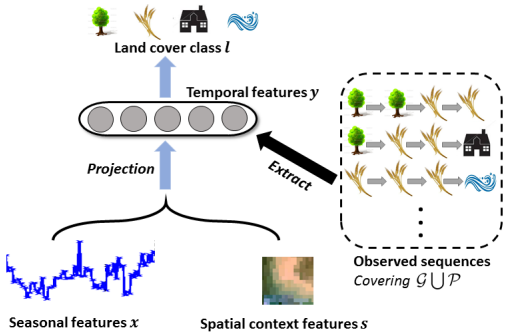


Figure 1: Zero-shot learning model: the temporal features of land cover classes are extracted from a set of observed land cover series. In prediction, the model first projects seasonal features and spatial context features to the temporal feature space, and then outputs the final land cover.

3.1 Zero-shot learning model

In land cover prediction problem, if we represent the set of land cover classes contained by ground-truth in the target region as $\mathcal{G}$ and the land cover classes that appear after T as $\mathcal{P}$, then it is possible that $\mathcal{P} \backslash \mathcal{G} \neq \emptyset$, where $\cdot \backslash \cdot$ denotes

the set difference. Therefore we propose a customized zero-shot learning model, which aims to recognize each land cover class in $\mathcal{G} \bigcup \mathcal{P}$ even without labelled training samples [23].

In traditional zero-shot learning tasks such as visual recognition, a semantic space is first created using auxiliary information sources, e.g. large text corpus [23], which is independent to the provided ground-truth in visual recognition. The auxiliary sources contain sufficient knowledge thus to generate the semantic features for both existing and unseen classes. Then the goal of zero-shot learning is to learn the projection from input image features to the semantic features using the training set.

In our problem, the raw spectral features are in high dimensional space and contain much noise, and also we do not have large text corpus as auxiliary information sources. To this end, we propose to extract three types of customized feature representation - seasonal features x, spatial context features s and temporal features y, and combine them into a zero-shot learning model, as depicted in Fig. 1. Here we consider a particular time step and omit the superscript t.

We extract seasonal features and spatial context features to summarize the spectral properties of each location. On the other hand, we learn the temporal features for each land cover class in $\mathcal{G} \bigcup \mathcal{P}$ using a set of observed land cover series, which serves as auxiliary information and will be described later. Our objective here is to learn the projection from the seasonal features and spatial context features of each location to the temporal features of the corresponding land cover class. After learning this projection using the ground-truth, we apply it to generate temporal features y for each test location. Then we take the closest land cover class to y in the temporal feature space as the final output. In this way we can identify both existing and unseen classes.

To illustrate the effectiveness of temporal features, we consider an example where the training set does not contain plantation locations. We assume two dimensions of the temporal features represent "whether this land cover is always converted from forest" and "whether this land cover will persist for a very long time". We have the knowledge from auxiliary sources that plantations have both these two properties, while the training set contains some land covers that have only one of these two properties. After learning the projection from input features to the temporal feature space using the training set, we can identify a plantation location if it reflects high values in these two dimensions. In practice, the extracted temporal features usually contain more abstract knowledge. We now present the involved feature representation in details.

Temporal features: To identify both existing and unseen land covers, we need to guarantee that 1) we can obtain the temporal features of unseen land covers, and 2) the temporal features contain semantics so that the projection learned from training set can also be applied on unseen land covers.

Due to these reasons, we learn a distributed representation for each land cover using the Continuous Bag-of-words (CBOW) model, which is an effective "word2vec" method in learning the semantic representation in natural language processing [16]. In our problem, we treat each land cover as a word, and $\mathcal{G} \bigcup \mathcal{P}$ forms the vocabulary. To get the "corpus", we collect multiple land cover series from different locations over years, e.g. "forest→cropland→cropland→wasteland". While the land covers in $\mathcal{P} \backslash \mathcal{G}$ do not exist in the target region, we can always collect land cover series from other places that contain these unseen land covers. In implementation, we collect land cover series from different places that contain all the land covers defined by land cover taxonomy [6].

Although these land cover series are collected from different places, they share similar transition patterns, and are then used to train CBOW. In this way we learn the distributed representation for each land cover. In the generated semantic space, each land cover label is represented by a continuous-valued vector. According to the theory of CBOW [16], two land covers with similar temporal transition patterns stay closer in the semantic space. Therefore, we name this generated representation as temporal features.

Seasonal features: The seasonal patterns are important in characterizing land covers. If we take the spectral features of a single date, it may be difficult to distinguish between a pair of land covers, e.g., a cropland just after harvest would look very similar to a barren land.

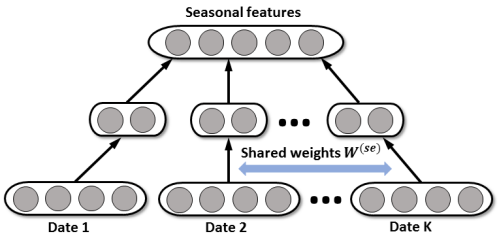


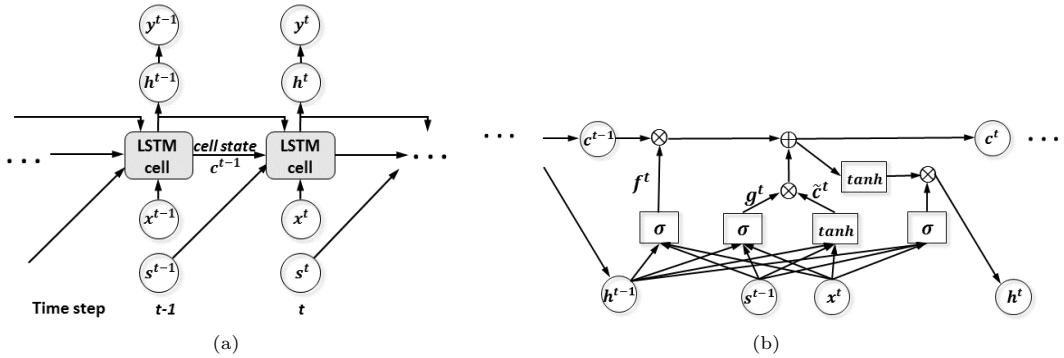
Figure 2: The two-layer neural networks which combines multiple dates to generates seasonal features.

For each location, we take the collected spectral features from multiple dates of a year and utilize two-layer neural networks to extract seasonal features, as shown in Fig. 2. The first layer is responsible for combining different dimensions (i.e. bandwidth) of multi-spectral data to generate discriminative features on each date via a weight matrix $W^{(se)}$. It is noteworthy that $W^{(se)}$ is shared among different dates. Then the extracted features from multiple dates are taken as input to the second layer, which combines the features from different dates via a weight matrix V. Specifically, this process at a time step/year t can be expressed as follows:

$$\begin{aligned} x^{t,(d)} &= \sigma(W^{(se)} z^{t,(d)}), \\ x^{t} &= \sigma(V[x^{(t,d=1:D)}]), \end{aligned} \tag{1}$$

where $z^{t,(d)}$ and $x^{t,(d)}$ denote the spectral features and the extracted features on each date d. $\sigma(\cdot)$ and $[\cdot]$ denote sigmoid function and concatenation function.

Spatial context features: We include spatial context features in the proposed model mainly for two reasons. First, the spatial contextual knowledge can provide useful insights



(a) (b)

Figure 3: (a) The variant of LSTM that includes spatial context features. (b) The structure of LSTM cell.

into land cover transitions. For instance, new croplands are usually cultivated around existing croplands. Second, since the locations of a specific land cover are usually contiguous over space, we can mitigate noisy spectral features by making classification consistent over space.

Compared with unsupervised extraction methods [10, 18], the extraction of spatial context features should be more related to the supervised information of land cover transitions, e.g. the locations showing "forest→ forest" and "forest→ cropland" should have different spatial context features.

In our extraction method, 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. To learn γ, we first define a probabilistic distribution similar to Neighborhood Component Analysis [3]. For each location i, it connects to another location j with a probability as:

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

where d_{ij} is the Euclidean distance between s_i^t and s_j^t.

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

$$q_{ij} = \frac{exp(-\rho_{ij}^2)}{\sum_{j'} exp(-\rho_{ij'}^2)}. \tag{3}$$

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

$$\rho_{ij} = \begin{cases} \infty, & l_i^t \neq l_j^t \,\&\, l_i^{t+1} \neq l_j^{t+1}, \\ max(\delta_i, \delta_j)\sqrt{(d_y^t)^2 + (d_y^{t+1})^2}, & otherwise, \end{cases} \tag{4}$$

where $d_y^t = ||y_i^t - y_j^t||$ is the Euclidean distance between the temporal features y_i^t and y_j^t, which corresponds to l_i^t and l_j^t. $\delta_i = p(l_i^{t+1}|l_i^t)$, measuring the fraction of locations with label l_i^t at time t to be converted to l_i^{t+1} at $t+1$, or the popularity of this transition. Since popular transitions are usually more interesting, in Eq. 4 we set a larger target distance between popular transitions and other transitions using δ_i and δ_j.

Our objective is to minimize the Kullback-Leibler (KL) divergence between our defined distribution p using s^t and the target distribution q. The gradient can be computed as:

$$\frac{\partial KL}{\partial \gamma} = \sum_i \frac{\partial KL}{\partial s_i^t}\frac{\partial s_i^t}{\partial \gamma} \propto -\sum_i \frac{\partial s_i^t}{\partial \gamma}\sum_j (s_i^t - s_j^t)(q_{ij} - p_{ij}). \tag{5}$$

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. In our implementation we cluster the data in each transition type and learn γ based on some sampled locations from each cluster.

3.2 Incremental Dual-memory LSTM

LSTM has shown extensive prospect in a variety of sequential labeling applications, including natural language processing and visual recognition [4, 27]. Compared with other sequential model such as Recurrent Neural Networks (RNN), the success of LSTM mainly stems from its capacity to model temporal dependencies over a long period [24].

In our problem, the spectral features of each land cover usually change over years, and also change in different stages, e.g. early plantations vs. mature plantations. Because of the temporal variation, the zero-shot projection from seasonal features and spatial context features to temporal features also changes. Hence, traditional LSTM cannot be used by itself to predict land covers. In this work, we extend LSTM with a dual-memory structure, which consists of long-term memory and short-term memory. The long-term memory is responsible for capturing long-term variation patterns from the long history, while the short-term memory captures the environmental change during more recent time steps. The dual-memory structure is incrementally updated to learn the latest knowledge about the ever-changing environment, and also facilitates the land cover prediction. We name our proposed framework as **I**ncremental **D**ual-memory **LSTM** (ID-LSTM).

We will first introduce a variant of LSTM model which learns the projection from seasonal features and spatial context features to temporal features. Then we further extend this model with a dual-memory structure. Finally, we will discuss the incremental update process.

3.2.1 Spatial LSTM. In this model, we wish to learn discriminative hidden knowledge to recognize our desired land covers, and also leverage temporal and spatial dependencies in land cover transitions. Therefore we introduce the hidden representation h^t, as shown in Fig. 3 (a). The input seasonal features x^t and the hidden representation h^t are connected via an LSTM cell. Besides, we include the temporal and spatial dependencies in LSTM cell to generate the hidden representation at next time step. In our problem, a target location is more likely to convert to certain land covers (e.g. cropland and burned area) at time step t 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 information at $t-1$ and include it to generate the hidden representation at t.

Here we briefly introduce the LSTM cell, as shown in Fig. 3 (b). 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. The cell state c^t is generated by combining c^{t-1} and the information at t. Hence 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 x^t, h^{t-1} and s^{t-1} into a $tanh(\cdot)$ function, as:

$$\tilde{c}^t = tanh(W_h^c h^{t-1} + W_x^c x^t + W_s^c s^{t-1}), \tag{6}$$

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 f^t and an input gate layer g^t are generated as follows:

$$\begin{aligned} f^t &= \sigma(W_h^f h^{t-1} + W_x^f x^t + W_s^f s^{t-1}), \\ g^t &= \sigma(W_h^g h^{t-1} + W_x^g x^t + W_s^g s^{t-1}), \end{aligned} \tag{7}$$

where $\{W_h^f, W_x^f, W_s^f\}$ and $\{W_h^g, W_x^g, W_s^g\}$ denote two sets of weight parameters for generating forget gate layer f^t and input gate layer g^t, respectively. 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:

$$c^t = f^t \otimes c^{t-1} + i^t \otimes \tilde{c}^t, \tag{8}$$

where $\otimes$ denotes entry-wise product.

Finally, we generate hidden representation by filtering the the obtained cell state using a output gate layer o^t, as:

$$\begin{aligned} o^t &= \sigma(W_h^o h^{t-1} + W_x^o x^t + W_s^o s^{t-1}), \\ h^t &= o^t \otimes tanh(c^t), \end{aligned} \tag{9}$$

where W_h^o, W_x^o and W_s^o are the weight parameters that are used to generate the hidden gate layer.

3.2.2 Dual-memory structure. The land cover transition has shown to be affected by both long-term cyclic events and short-term environmental changes. Therefore we extend previous LSTM model using the dual-memory structure, as shown in Fig. 4. The dual-memory structure consists of long-term memory and short-term memory, and they

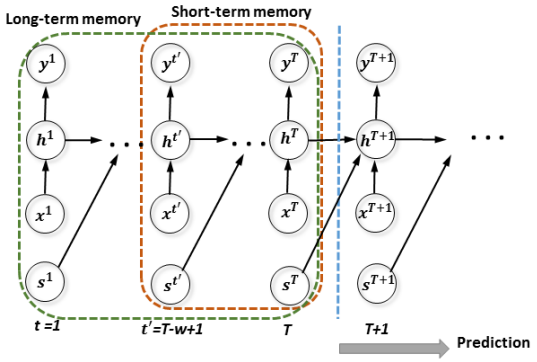


Figure 4: The dual-memory LSTM model. Here we represent dependencies between cell states by the dependencies between hidden states for simplicity.

both contain a set of samples in their respective time windows. Considering the prediction at time step $t+1$, since the long-term memory is responsible for memorizing the pattern from long history, its time window covers time steps from 1 to t. In contrast, the short-term memory captures the recent variation patterns, and its time window covers most recent w time steps before prediction, i.e. from $t-w+1$ to t.

Based on the introduced dual-memory structure, we maintain two sets of cell states over time, $\{c_{(l)}^t\}$ and $\{c_{(s)}^t\}$, to store long-term information and short-term information. The cell states $c_{(l)}$ and $c_{(s)}$ have separate model parameters $\theta_{(l)} = \{W_{(l)}^c, W_{(l)}^f, W_{(l)}^g\}$ and $\theta_{(s)} = \{W_{(s)}^c, W_{(s)}^f, W_{(s)}^g\}$, but have shared weights W^o. Here we use the notation $W^\star = \{W_x^\star, W_s^\star, W_h^\star\}$, $\star = c/f/g/o$ for simplicity. The parameters $\theta_{(l)}$ and $\theta_{(s)}$ are learned using the samples in long-term memory and short-term memory, respectively. Then during the prediction, we generate a new variable η^t via parameters $\{P_x, P_s\}$ to determine which memory is used to generate h^t. In real-world applications, some locations tend to follow long-term variation patterns, while some other locations follow short-term environmental changes, depending on their spectral properties and surrounding environment. Using η^t to summarize these factors, the generative process can be described as follows:

$$\begin{aligned} \eta^t &= \sigma(P_x x^t + P_s s^{t-1}) \\ h^t &= \begin{cases} o^t \otimes tanh(c_{(l)}^t), \eta^t > 0.5, \\ o^t \otimes tanh(c_{(s)}^t), otherwise, \end{cases} \end{aligned} \tag{10}$$

Then we compute the temporal features y^t and the label l^t by weight parameter U, as:

$$\begin{aligned} y^t &= Uh^t, \\ l^t &\leftarrow CL(y^t). \end{aligned} \tag{11}$$

where $CL(\cdot)$ denotes the operation of selecting the closest land cover class to y^t in the temporal feature space. The involved parameters in the entire framework (including the parameters to generate seasonal features, the parameters in

LSTM and the parameters in generating η^t) can be inferred using back-propagation algorithm.

3.2.3 Incremental Update. During the prediction process from $T+1$ to $T+m$, the temporal variation will degrade the classification performance as time progresses. Besides, since the unseen classes are not included in the initial training set, applying the learned projection from existing classes on the unseen classes may cause shift/bias, which is also referred to as projection shift problem [2]. For these reasons, we propose to incrementally update parameters and temporal features of land covers to adapt the learning framework to the ever-changing environment and refine the zero-shot projection.

The whole update process can be implemented in a recursive EM-style process. Here we consider the prediction at $t+1$, $t = T$ to $T+m-1$. In E-step we estimate h^{t+1} from the information at t by Eq. 10, and assign the label $\hat{l}^{t+1}$ for each location by Eq. 11. Here we use the notation $\hat{l}$ to distinguish the predicted label from the provided label l.

Then in M-step, we move the time windows of long-term memory and short-term memory. After this move, the long-term memory covers the time steps from 1 to $t+1$, and the short-term memory covers the time steps from $t-w+2$ to $t+1$. Then we update the parameters $\theta_{(l)}$ and $\theta_{(s)}$ using the samples in respective time windows, as follows:

$$\begin{aligned}\theta_{(l)}^{new} &= argmin_{\theta_{(l)}} L(y^{1:t+1}, \{v(l^{1:T}), v(\hat{l}^{T+1:t+1})\}),\\ \theta_{(s)}^{new} &= argmin_{\theta_{(s)}} L(y^{t-w+2:t+1}, \{v(l^{t-w+2:T}), v(\hat{l}^{T+1:t+1})\}),\end{aligned} \tag{12}$$

where $\{\cdot,\cdot\}$ denotes the union of two sets of vectors, $L(\cdot)$ is the squared loss function, and $v(l)$ denotes the temporal features associated with label l. Intuitively, we wish to minimize the difference between the predicted temporal features y and the temporal features associated with the obtained land cover labels in each time window.

After updating model parameters, we also refine the temporal features of each land cover. Specifically, we update the temporal features $v(l)$ of each land cover l as the centroid of the predicted temporal features of all the locations in land cover l at $t+1$. Such update process can not only alleviate temporal variation, but also refine the projection to the unseen classes that appear during the prediction process.

As summarized in Algorithm 1, the incremental prediction process has a time cost of $O(\kappa mN)$, where κ is a constant determined by the dimensionality of our customized feature representation and hidden representation.

Algorithm 1 Incremental learning in prediction.

Input: $\{z^1, ..., z^T, z^{T+1}, ..., z^{T+m}\}$: A series of spectral features; The learned model before T.
Output: $\{\hat{l}^{T+1:T+m}\}$
1: **for** time step $t \leftarrow T$ to $T+m-1$ **do**
2: Generate x^{t+1} and s^t.
3: Estimate $\hat{l}^{t+1}$ by Eqs. 10 and 11.
4: Move the time windows, include $\hat{l}^{t+1}$ as training labels.
5: Update $\theta_{(l)}$ and $\theta_{(s)}$ by Eq. 12.
6: Infer y^{t+1} and $\hat{l}^{t+1}$ for all locations at time $t+1$.
7: Update the temporal features $v(l)$ as the centroid of inferred y^{t+1} of each land cover l.
8: **end for**

4 EXPERIMENTS

In this section we present a detailed evaluation and reasoning behind the results of our proposed method. We first introduce the baselines in our tests:

Artificial Neural Networks (ANN): In this baseline we train a global ANN model using raw spectral features and provided labels from all the time steps.

Recurrent Neural Networks (RNN): We train an RNN model using raw spectral features and labels.

Long short-term memory (LSTM): Similarly, we train an LSTM model using raw spectral features and labels.

Spatial LSTM (sLSTM): We combine seasonal features and spatial context features in LSTM (Fig. 3 (a)).

Incremental long-term memory (ilLSTM): Based on sLSTM, we conduct incremental learning using only long-term memory. The comparison between ilLSTM and ID-LSTM can reveal the effectiveness of dual-memory structure.

We evaluate our method with respect to all the baselines in both synthetic dataset and real-world applications. For all of these evaluations, we utilize 500-meter resolution MODIS multi-spectral product MOD09A1 as input spectral features. MOD09A1 product contains multi-spectral data with 7 reflectance bands (620-2155 nm) collected by MODIS instruments onboard NASA's satellites. In this product, 8-day composite images are generated from daily images by selecting the per-pixel reflectance value with least disturbances (i.e. clouds and missing values) from every 8-day interval. For each year, we take 15 most discriminative images out of 46 total composite images available for the entire year according to domain knowledge (e.g. land covers are less distinguishable during cloudy and winter seasons). In our experiments, we define $N(i)$ to be the set of locations within a 1500m by 1500m squared range centered at i^{th} location.

4.1 Synthetic Dataset

We first evaluate our proposed on a synthetic dataset. Specifically, we create a virtual region which gradually changes over 20 time steps, as shown in Fig. 5. The created region contains five types of land covers: forest, cropland, urban area, water body and wasteland. For the first 14 time steps, the spectral features of each location are the real spectral features (from MODIS) of randomly selected locations in each land cover during 2001-2014. For time step 15-20, we train an LSTM model to generate spectral features at each time step using the spectral features at previous time step.

We evaluate our proposed method and the baselines in two tests. In Test 1, we train each model using the first 10 time steps, and conduct prediction on the next 10 time steps. It is noteworthy that the class of "urban area" does not appear in the training set. In Test 2, we train each model using the first 15 time steps, and then test on the next 5 time steps.

Table 1: Performance (F1-score on urban area and cropland) in Test 1 at time step 11-20 on synthetic data.

Method	Class	11	12	13	14	15	16	17	18	19	20
ANN	urban	0.773	0.764	0.743	0.726	0.701	0.702	0.688	0.677	0.673	0.664
	crop	0.825	0.820	0.812	0.808	0.776	0.758	0.757	0.754	0.733	0.716
RNN	urban	0.788	0.769	0.753	0.752	0.738	0.723	0.725	0.721	0.716	0.704
	crop	0.838	0.832	0.825	0.815	0.791	0.790	0.788	0.782	0.748	0.748
LSTM	urban	0.802	0.778	0.762	0.752	0.757	0.746	0.737	0.732	0.728	0.719
	crop	0.853	0.850	0.837	0.828	0.817	0.802	0.790	0.764	0.767	0.753
sLSTM	urban	0.818	0.792	0.779	0.766	0.761	0.755	0.749	0.740	0.733	0.728
	crop	0.873	0.868	0.851	0.844	0.830	0.815	0.808	0.784	0.773	0.762
ilLSTM	urban	0.818	0.812	0.807	0.789	0.779	0.774	0.758	0.753	0.742	0.740
	crop	0.873	0.871	0.864	0.865	0.852	0.844	0.822	0.810	0.804	0.783
ID-LSTM	urban	0.852	0.858	0.850	0.833	0.807	0.785	0.769	0.766	0.753	0.757
	crop	0.912	0.906	0.902	0.897	0.872	0.865	0.848	0.836	0.819	0.805

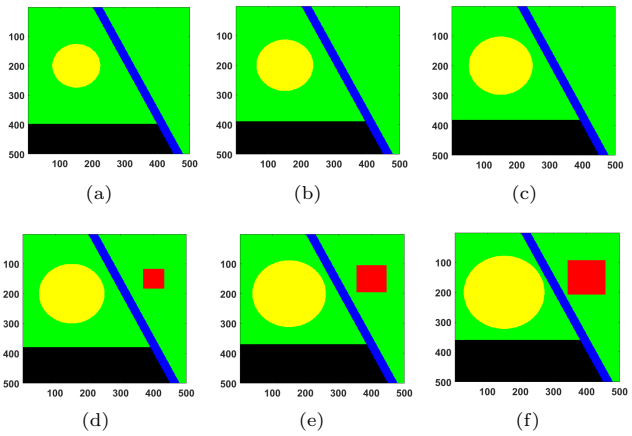


Figure 5: The created synthetic dataset at time step 1(a), 5(b), 10(c), 11(d), 15(e), 20(f). Color legend: yellow - cropland, green - forest, blue - water body, black - wasteland, red - urban area.

To populate the temporal features of each land cover type, we randomly collect land cover series from 100,000 locations from Southeast Asia from 2001 to 2014. In all the following tests we set the window size $w = 4$ according to domain knowledge and the performance on a validation set.

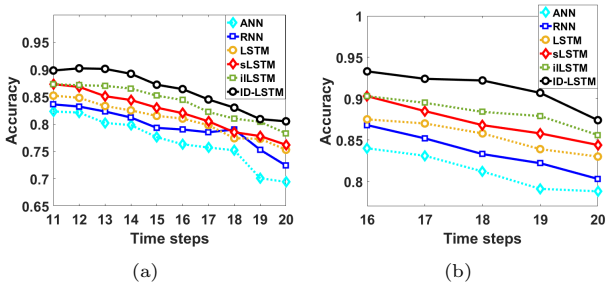


Figure 6: The prediction performance (accuracy) in (a) Test 1 during 11-20 and (b) Test 2 during 16-20.

We show the classification performance in two tests in Fig. 6 (a) and (b), respectively. It is clear that the classification accuracy of all the methods decreases over time due to the temporal variation. However, we can observe that ID-LSTM and ilLSTM decrease much slower than the other methods, especially at the first several time steps. As time progresses, the accumulated temporal variation results in poor performance for all the methods. Compared to ilLSTM, the improvement made by ID-LSTM mainly stems from its capacity in capturing recent variation patterns.

Since the test data is skewed among land covers, we also measure the performance using F1-score on urban area and cropland. According to Table 1, ID-LSTM outperforms other methods in predicting both classes (∼13% improvement over ANN). Moreover, we note that the performance on cropland is much better than that on urban area. As urban area does not appear in training set, the learned projection may not be accurate on urban area due to the projection shift. However, the results show that ilLSTM and ID-LSTM, by utilizing incremental update on model parameters and temporal features, lead to much better performance in predicting urban areas, especially at first several time steps.

4.2 Oil Palm Plantation Detection

Here we validate our framework in detecting oil palm plantations, which is a key driver for deforestation in Indonesia. Since plantations have similar properties (e.g. greenness) with tropical forest, most products are manually created.

Table 2: The plantation detection performance of each method in 2009-2014, measured in F1-score.

Method	Set	2009	2010	2011	2012	2013	2014
ANN	train	0.775	0.732	0.708	0.657	0.661	0.634
	test	0.760	0.684	0.662	0.650	0.601	0.603
RNN	train	0.776	0.754	0.720	0.695	0.660	0.641
	test	0.761	0.703	0.654	0.646	0.621	0.618
LSTM	train	0.794	0.779	0.748	0.736	0.685	0.653
	test	0.768	0.712	0.680	0.653	0.640	0.624
sLSTM	train	0.841	0.808	0.786	0.736	0.709	0.704
	test	0.828	0.746	0.744	0.740	0.700	0.664
ilLSTM	train	0.841	0.816	0.796	0.773	0.743	0.742
	test	0.828	0.799	0.773	0.756	0.723	0.721
ID-LSTM	train	0.863	0.842	0.861	0.826	0.820	0.826
	test	0.846	0.830	0.834	0.822	0.809	0.805

We utilize two latest manually created datasets - RSPO [6] and Tree Plantation [20] to create ground-truth. RSPO is available in 2000, 2005, and 2009 while Tree Plantation is only available in 2014. We combine both datasets and utilize Enhanced Vegetation Index (EVI) time series from 2001 to 2014 to create yearly ground-truth for 50,000 locations in Kalimantan, Indonesia through 2001-2014. Each location is labeled as one of the categories from {cropland, urban area, disturbed forest, undisturbed forest, mining, palm oil plantation, timber plantation, wasteland, water body} according to the land cover taxonomy in [6].

We train each method with the ground-truth on 25,000 randomly selected locations before 2008, and then test on both training locations and the remaining 25,000 test locations for each year from 2009 to 2014. Since our created ground-truth is more accurate on plantations than other classes, we measure the performance using the F1-score on plantation class.

According to Table 2, ID-LSTM outperforms other methods in detecting plantations by a considerable margin. The prediction by ANN is unsatisfactory mainly due to its ignorance of spatio-temporal dependencies. Besides, the comparison between LSTM and sLSTM shows the effectiveness of including seasonal features and spatial context features in classification. The improvement from ilLSTM to ID-LSTM stems from the capacity of dual-memory structure in better modeling the locations that are impacted by short-term variation patterns. Finally, we conclude that the incremental update is critical for addressing temporal variation.

Table 3: The plantation prediction performance in 2009-2014, measured in F1-score. Each method is trained using the ground-truth during 2001-2005.

Method	2009	2010	2011	2012	2013	2014
ANN	0.613	0.610	0.603	0.610	0.582	0.568
RNN	0.650	0.641	0.628	0.623	0.611	0.602
LSTM	0.695	0.685	0.673	0.674	0.645	0.613
sLSTM	0.722	0.714	0.708	0.684	0.672	0.637
ilLSTM	0.741	0.742	0.730	0.682	0.678	0.658
ID-LSTM	0.789	0.776	0.754	0.756	0.736	0.703

Although palm oil plantations expand very fast, there exist few plantations in our study region before 2005. We conduct another test using 30,000 selected locations, none of which are plantations before 2005. We train each model using the ground-truth before 2005, and detect plantations after 2005. Due to the space limit, here we only show the performance after 2008 in Table 3. We observe that the performance greatly drops compared to the values in Table 2 due to the projection shift. However, we can still observe the superiority of ID-LSTM over other methods.

Given the previous results, we wish to better understand the impact of accumulated temporal variation over years. Compared with non-plantation locations, true plantation locations are found to be closer to plantation class in temporal feature space. However, the distance gradually increases over years so they may be misclassified. In Fig. 7 we show

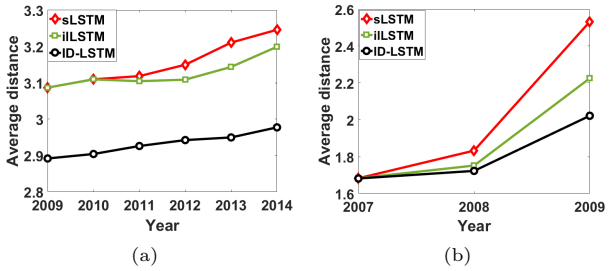


Figure 7: (a) The average distance from true plantation locations to "plantation" class in temporal feature space from 2009-2014. (b) The average distance from burned locations to "burned area" class in temporal feature space from 2007-2009.

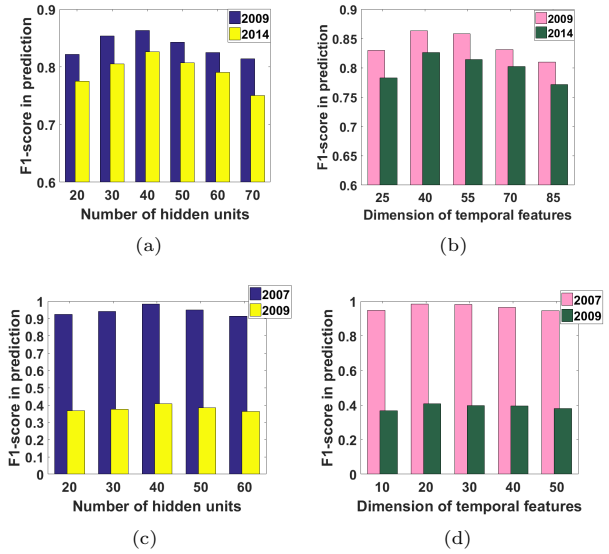
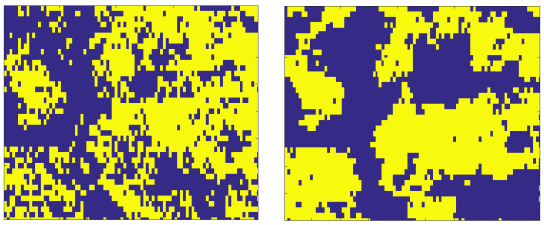


Figure 8: The plantation prediction performance (F1-score) with different dimensionality of (a) hidden representation, and (b) temporal features. The performance (F1-score) on burned area under different dimensionality of (c) number of hidden variables, and (d) dimension of temporal features.

the progression of average distance from true plantation locations to the temporal features of plantation class. We conclude that ID-LSTM is very effective in maintaining low distance values. The comparison between sLSTM and ilLSTM demonstrates the effectiveness of incremental update.

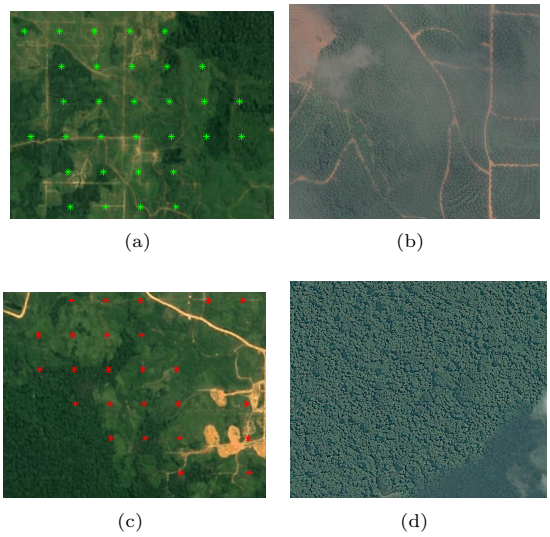
Furthermore, we conduct parameter sensitivity test. In Fig. 8 (a) and (b), we measure the performance by changing dimensionality of hidden representation and temporal features. As observed, the dimensionality needs to be carefully chosen to reach the balance between bias and variance.

In Fig. 9, we show detected plantations in a specific 40×90 region using ANN and ID-LSTM. We can see that ID-LSTM can better delineate the plantation boundary and the detected plantations are contiguous over space while ANN results in both false positives and false negatives. We then compare the generated product by ID-LSTM to the manually created plantation ground-truth. In Fig. 10 (a), we show a region which is detected by our method but missed by ground-truth. In contrast, Fig. 10 (c) shows a region detected by ground-truth, but classified as disturbed forest by ID-LSTM. The high-resolution images in Fig. 10 (b) and (d) confirms that ID-LSTM generates correct results in both cases. Therefore we conclude that our proposed framework can ensure the high quality of the generated land cover mapping product.



(a) (b)

Figure 9: The detected plantation locations (yellow) in a test region using (a) ANN and (b) ID-LSTM.



(a) (b)

(c) (d)

Figure 10: (a) A plantation region detected by ID-LSTM (c) A forest region mistakenly detected by ground-truth. (b)(d) The high-resolution validation images (DigitalGlobe) corresponding to (a) and (c).

4.3 Identifying Burned Area

Besides the planation detection, we also evaluate our method on another application - identifying burned areas in Montana state, US. This task does not involve any novel/unseen classes, but is challenged by strong temporal variation and the seasonal patterns of forest fires. We obtained fire validation data until 2009 from government agencies responsible for monitoring and managing forests and wildfires. In total we select 15,107 MODIS locations and each location has a label from {burned area, forest, other} in every year from 2001 to 2009. In this application, the "other" class contains multiple types of land covers and these locations have different degrees of temporal variation.

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 our customized feature representation and the incremental update of dual-memory structure bring considerable improvement. On the other hand, we can find that all the methods have low F1-scores in 2009. This result is due to the fact that the burned locations in 2009 are very few, and both precision and recall will be severely disturbed by any classification errors.

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

Method	Set	2007	2008	2009
ANN	train	0.840	0.554	0.116
	test	0.774	0.482	0.073
RNN	train	0.895	0.618	0.124
	test	0.868	0.582	0.079
LSTM	train	0.905	0.643	0.163
	test	0.902	0.619	0.150
sLSTM	train	0.981	0.834	0.258
	test	0.976	0.806	0.244
liLSTM	train	0.981	0.907	0.363
	test	0.976	0.894	0.346
ID-LSTM	train	0.984	0.943	0.407
	test	0.978	0.939	0.368

We then conduct sensitivity test and obtain similar results, as depicted in Fig. 8 (c) and (d). We also track the average distance of all the burned locations to the "burned area" class in the temporal feature space. According to Fig. 7 (b), the average distance by ID-LSTM increases much slower than the other methods.

5 RELATED WORK

Discovering LULC changes is essential for understanding environmental change [12, 21]. Recent advances in collecting remote sensing data have spawned much research on monitoring land cover in large regions [1, 8]However, there are still many challenges in identifying certain land covers. For e.g., detecting burned area is difficult since fires have a seasonal pattern and only last for a few months.

Land cover prediction becomes even more challenging due to the emergence of unseen land cover classes and temporal variation [9, 11]. Although there exist research works that

utilize zero-shot learning [2, 23] in identifying unseen classes, most of them focus on natural language processing and do not tackle the projection shift. In [14], a dual-memory model is used to capture the variation in streaming data, but cannot be directly applied in land cover problem.

Conventional machine learning models have been widely explored in a variety of land cover prediction problems [7, 17, 22, 25]. However, these methods have limited capacity to extract discriminative information and capture spatio-temporal relationship from large amount of remotely sensed data. While deep learning models such as RNN, LSTM and word2vec have shown promising performance in sequential data mining [4, 13, 15, 26], their application in land cover discovery is still limited. Most of these methods do not make fully use of spatio-temporal information in modeling land cover transitions. When used in land cover prediction, they are also vulnerable to temporal variation and noisy spectral features.

6 ACKNOWLEDGEMENT

This work was funded by the NSF Awards 1029711, and Gordon and Betty Moore Foundation and the Belmont Forum/FACCE-JPI funded DEVIL project (NE/M021327/1). Access to computing facilities was provided by NASA Earth Exchange and Minnesota Supercomputing Institute.

7 CONCLUSION

In this paper, we propose ID-LSTM for land cover prediction. Compared with traditional classification methods, ID-LSTM contains two memories that can capture both long-term and short-term variation patterns. The dual-memory structure is incrementally updated to include the latest information about the ever-changing environment. Experiments on both synthetic real-world datasets demonstrate that ID-LSTM can successfully detect both existing and unseen classes. Also, it is observed that the incremental update of dual-memory structure can effectively address the temporal variation. In addition, our comparison with state-of-the-art plantation ground-truth data shows that ID-LSTM can generate a high-quality product, and thus has a potential to contribute to a larger community of land cover problems and to assist in understanding global environmental change.

REFERENCES

[1] Kimberly M Carlson, Lisa M Curran, Gregory P Asner, Alice McDonald Pittman, Simon N Trigg, and J Marion Adeney. 2013. Carbon emissions from forest conversion by Kalimantan oil palm plantations. *Nature Climate Change* (2013).

[2] Yanwei Fu, Timothy M Hospedales, Tao Xiang, Zhenyong Fu, and Shaogang Gong. 2014. Transductive multi-view embedding for zero-shot recognition and annotation. In *European Conference on Computer Vision*. Springer.

[3] Jacob Goldberger, Geoffrey E Hinton, Sam T Roweis, and Ruslan Salakhutdinov. 2004. Neighbourhood components analysis. In *NIPS*.

[4] A Graves, M Liwicki, S Fernndez, R Bertolami, H Bunke, and J Schmidhuber. 2009. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2009).

[5] Nancy B Grimm, Stanley H Faeth, Nancy E Golubiewski, Charles L Redman, Jianguo Wu, Xuemei Bai, and John M Briggs. 2008. Global change and the ecology of cities. *science* (2008).

[6] P. Gunarso, M. E. Hartoyo, F. Agus, Killeen, T. J., and J. Goon. 2013. RSPO, Kuala Lumpur, Malaysia. *Reports from the technical panels of the 2nd greenhouse gas working group of the Roundtable on sustainable palm oil* (2013).

[7] Collin Homer, Chengquan Huang, Limin Yang, Bruce Wylie, and Michael Coan. 2004. Development of a 2001 national land-cover database for the United States. *Photogrammetric Engineering and Remote Sensing* (2004).

[8] Xiaowei Jia, Ankush Khandelwal, James Gerber, Kimberly Carlson, Paul West, and Vipin Kumar. 2016. Learning large-scale plantation mapping from imperfect annotators. In *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE.

[9] Xiaowei Jia, Ankush Khandelwal, Guruprasad Nayak, James Gerber, Kimberly Carlson, Paul West, and Vipin Kumar. 2017. Predict Land Covers with Transition. Modeling and Incremental Learning. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM.

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

[11] Anuj Karpatne, Zhe Jiang, Ranga Raju Vatsavai, Shashi Shekhar, and Vipin Kumar. 2016. Monitoring Land-Cover Changes: A Machine-Learning Perspective. *IEEE Geoscience and Remote Sensing Magazine* (2016).

[12] Eric F Lambin, Bi L Turner, Helmut J Geist, Samuel B Agbola, Arild Angelsen, John W Bruce, Oliver T Coomes, Rodolfo Dirzo, Günther Fischer, Carl Folke, and others. 2001. The causes of land-use and land-cover change: moving beyond the myths. *Global environmental change* (2001).

[13] Xiaoyi Li, Xiaowei Jia, Hui Li, Houping Xiao, Jing Gao, and Aidong Zhang. 2015. DRN: Bringing Greedy Layer-Wise Training into Time Dimension. In *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE.

[14] Viktor Losing, Barbara Hammer, and Heiko Wersing. 2016. KNN Classifier with Self Adjusting Memory for Heterogeneous Concept Drift. In *ICDM*.

[15] Haobo Lyu, Hui Lu, and Lichao Mou. 2016. Learning a Transferable Change Rule from a Recurrent Neural Network for Land Cover Change Detection. *Remote Sensing* (2016).

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[17] Guruprasad Nayak, Varun Mithal, and Vipin Kumar. 2016. Multiple Instance Learning for burned area mapping using multi - temporal reflectance data. (2016).

[18] Mohammad Norouzi, Mani Ranjbar, and Greg Mori. 2009. Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning. In *CVPR*.

[19] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. 2009. Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*.

[20] RACHAEL Petersen, ED Goldman, N Harris, S Sargent, D Aksenov, A Manisha, and others. 2016. Mapping tree plantations with multispectral imagery: preliminary results for seven tropical countries. *WRI* (2016).

[21] Roger A Pielke. 2005. Land use and climate change. In *Science*.

[22] Tobias Sauter, Björn Weitzenkamp, and Christoph Schneider. 2010. 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).

[23] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*.

[24] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM Neural Networks for Language Modeling.. In *Interspeech*.

[25] Qiong Wu, Hong-qing Li, Ru-song Wang, Juergen Paulussen, Yong He, Min Wang, Bi-hui Wang, and Zhen Wang. 2006. Monitoring and predicting land use change in Beijing using remote sensing and GIS. *Landscape and urban planning* (2006).

[26] Guangxu Xun, Xiaowei Jia, Vishrawas Gopalakrishnan, and Aidong Zhang. 2016. A Survey on Context Learning. *IEEE Transactions on Knowledge and Data Engineering* (2016).

[27] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. 2015. Beyond short snippets: Deep networks for video classification. In *IEEE CVPR*.