# A Relative Density Ratio-Based Framework for Detection of Land Cover Changes in MODIS NDVI Time Series

Asim Anees, Jagannath Aryal, Małgorzata M. O'Reilly, and Timothy J. Gale

***Abstract*—To improve statistical approaches for near real-time land cover change detection in nonGaussian time-series data, we propose a supervised land cover change detection framework in which a MODIS NDVI time series is modeled as a triply modulated cosine function using the extended Kalman filter and the trend parameter of the triply modulated cosine function is used to derive repeated sequential probability ratio test (RSPRT) statistics. The statistics are based on relative density ratios estimated directly from the training set by a relative unconstrained least squares importance Fitting (RULSIF) algorithm, unlike traditional likelihood ratio-based test statistics. We test the framework on simulated, synthetic, and real-world beetle infestation datasets, and show that using estimated relative density ratios, instead of assuming the individual density functions to be Gaussian or approximating them with Gaussian Kernels, in the RSPRT statistics achieves better performance in terms of accuracy and detection delay. We verify the efficiency of the proposed approach by comparing its performance with three existing methods on all the three datasets under consideration in this study. We also propose a simple heuristic technique that tunes the threshold efficiently in difficult cases of near real-time change detection, when we need to take three performance indices, namely, false positives, false negatives, and mean detection delay, into account simultaneously.**

***Index Terms*—Change detection, extended Kalman filter (EKF), model fitting, MODIS, relative density ratio, time series.**

## I. Introduction

LAND COVER change detection research has seen significant recent contributions [1]–[19]. However, every proposed framework has its limitations on global scale due to particularity of the task at hand and the circumstances under which it is developed. Hence, no single framework is optimal in a wide range of scenarios simultaneously. Therefore, an efficient change detection framework is always in demand for a particular task and circumstances under consideration. In this study, our main focus is on the changes induced by beetle infestations in pine forests, which is one of the major causes of land cover changes in North America [2], [3]. We utilize coarse spatial resolution MODIS data due to its free-of-cost availability and high temporal resolution.

Many studies have been published which utilize coarse spatial resolution data in addressing land cover change detection [10]–[18], [20]. Some of these studies propose methods which are not designed for detecting changes sequentially in near real time [11], [12], [15], [17], [18], [20]. In order to be able to mitigate the factors that are causing unwanted changes, early detection is crucial [10], [14], [16]. Therefore, considerable importance has been given to statistical approaches for near real-time land cover change detection over the recent past [10], [13], [14], [16], [21], [22].

The term "near real time" theoretically means that the algorithm can detect a change event with a small delay (in terms of number of observations) after the time point at which it has actually occurred, using only current and past observations. However, in remote sensing, its meaning is relative, depending on the type of application, i.e., types of changes being targeted and the data being used. Some changes are gradual and any change detection method may take a considerable number of observations before detecting them, still the methods are termed as near real time. For example, the studies published in [10], [14], and [16] introduced near real-time methods for detecting changes in MODIS time-series data. The number of observations required by these methods before detecting changes may sound nonreal time, but because of the type of changes addressed in these studies, i.e., beetle infestations (slow and gradual), they are declared as near real time.

Some existing methods [13], [20] derive test statistics from the raw vegetation index time series or the error between the model and observed time series [10], but it was suggested in [14] and [16] that calculating change metrics from the parameter time series (time-varying parameters of the fitted model) achieve better performance. There are two main issues when we consider these statistical approaches for near real-time land cover change detection. First, most of them either assume that the underlying density functions under null or alternative hypothesis are Gaussian [10], [14], [16], or they use Gaussian kernels to estimate the individual density functions [13]. The real-world data may often be far from being Gaussian. This results in errors in estimated or assumed underlying distributions, which achieve suboptimal performance [23], [24].

Manuscript received February 11, 2015; revised April 17, 2015; accepted April 26, 2015.

A. Anees is with the School of Engineering and ICT, University of Tasmania, Hobart, TAS 7001, Australia (e-mail: Asim.Anees@utas.edu.au).

J. Aryal is with the Discipline of Geography and Spatial Sciences, School of Land and Food, University of Tasmania, Hobart, TAS 7001, Australia (e-mail: Jagannath.Aryal@utas.edu.au).

M. M. O'Reilly is with the School of Physical Sciences, University of Tasmania, Hobart, TAS 7001, Australia.

T. J. Gale is with the School of Engineering and ICT, University of Tasmania, Hobart, TAS 7001, Australia.

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

Digital Object Identifier 10.1109/JSTARS.2015.2428306

Second, commonly two performance indices, namely: false positives (FP) and false negatives (FN), are considered while evaluating the change detection methods [13]. Evaluation of a near real-time method can be based on three performance indices, namely FP, FN, and mean detection delay (MD), instead of only FP and FN as in the case of offline change detection methods [13], [14]. Increasing threshold normally reduces FP rate, but increases MD and FN, and vice versa. Therefore, finding an optimal tradeoff manually between acceptable values of the three performance indices becomes challenging. Note that the readers should not confuse MD with the computational time or computational complexity of the algorithm here. MD is the average detection delay calculated in terms of the number of time points (observations) between the actual point of change in the ground truth data (or selected reference point) and the point where change alarm is raised. Therefore, its units depends upon the temporal resolution of the data under consideration, which may vary from application to application. We use generic units (observations or time points) for MD in this manuscript, which can be interpreted easily in different applications, according to the temporal resolution of the time-series data being used.

It was argued in [23]–[27] that approximating individual densities or assuming the individual densities to be Gaussian leads to more errors. However, when the test statistics are based on density ratios, we can avoid estimating individual densities or assuming them to be Gaussian as shown in [23], [24]. It was shown in [23], [25], and [26] that direct density ratio estimation performs better than estimating individual densities using Gaussian Kernels or assuming them to follow Gaussanity. A number of studies have proposed different methods for direct density ratio estimation, e.g., kernel mean matching [28], the logistic regression method [29], and Kullback–Leibler importance estimation procedure (KLIEP) [25]. KLIEP was shown to be promising in change detection framework [23]. A more recent algorithm in this regard, namely unconstrained least squares importance fitting (ULSIF), was proposed in [26] and shown to have optimal nonparametric convergence rate [30], optimal numerical stability [31], and higher robustness than KLIEP [32]. However, [27] reported a potential weakness of density ratio-based approaches that density ratios can be unbounded, and proposed relative ULSIF (RULSIF) algorithm, which uses relative density ratios that are always bounded. RULSIF was shown to achieve better estimation and nonparametric convergence than ULSIF. Although the existing remote-sensing literature on land cover change detection contains several methods based on traditional likelihood ratios with individual densities either assumed to be Gaussian, or estimated using Gaussian kernels, there is no study to our knowledge that has exploited the usefulness of relative density ratio estimation as proposed in [24], [27].

In this study, one of our main aims is to highlight and re-emphasize the usefulness of relative density ratios in remote-sensing applications. We investigate the advantages of using relative density ratio estimation in supervised near real-time classification of change and no-change events within the MODIS NDVI time series. Our proposed change detection framework models MODIS 8-days 500-m NDVI time series by a triply modulated cosine function [17], [18] and uses the extended Kalman filter (EKF) [33] to derive its time-varying parameters. As suggested in [1], [14], and [15], changes which affect the trend of the signal can be captured in the trend parameter of the triply modulated cosine function. Therefore, our proposed framework learns relative density ratios from the trend parameters of the change and no-change training sets using the RULSIF algorithm [24]. Once the training is done, these estimated relative density ratios are used to derive repeated sequential probability (RSPRT) [33] statistics online, which can be compared to a tuned threshold to detect changes in near real time.

We also address the issue of finding an acceptable tradeoff between FP, FN, and MD while tuning the threshold. Traditionally, performance of the any change detection method is analyzed based only on accuracy. In such cases, the threshold tuning can be done with the help of receiver operating characteristics (ROC) curve or calibration curve [34]. The ROC curve helps in finding a tradeoff between TP and FP, and the calibration curve plots accuracy against different values of threshold. However, in our case, we have to consider MD as well, along with accuracy, which means that the optimal tradeoff has to be found between three performance indices, namely FN (instead of TP), FP, and MD. We formulate a cost function depending on FP, FN, MD (or alternatively kappa-coefficient [35] and MD) and the threshold, which is minimized iteratively to find a threshold value that gives acceptable tradeoff between these performance indices. Finally, we compare our proposed framework with three recently published land cover change detection methods for land cover change detection in MODIS NDVI time-series data which use either KDE to estimate the individual densities or assume them to be Gaussian, while deriving the test statistics [13]. We show that our proposed framework achieves better accuracy with lower detection delays.

The main contributions of this study are as follows: 1) a supervised near real-time change detection framework that can detect land cover changes in MODIS NDVI time-series data quicker and with more accuracy than recently published methods; 2) highlighting and re-emphasizing the usefulness of density/relative density ratio estimation [23], [24], [27] in the remote-sensing community while showing its suitability for supervised change detection in MODIS time-series data; and 3) an effective strategy for tuning the threshold automatically in near real-time scenarios when more than two performance indices have to be considered, and also in those scenarios where manual threshold selection is cumbersome, e.g., cross-validation experiments.

The research questions addressed in this study are: 1) Is the relative density ratio estimation a viable option for supervised change detection in MODIS time series data? 2) Do RSPRT/CUSUM (CUmulative SUM) statistics [33], [36], [37], when derived from the parameter time series, improve performance compared to when derived from the raw time series [13]? 3) Does using the relative density ratios, estimated by RULSIF [23], [24], [27] in RSPRT statistics improve the performance compared to estimating the individual densities [13] or assuming them to be Gaussian [10], [14]?

This paper is organized as follows. Section II explains the RULSIF algorithm for relative density ratio estimation, the

proposed supervised land cover change detection framework, and the proposed threshold tuning technique. Section III gives brief descriptions of the three existing methods used in this study for comparison and performance evaluation of our proposed framework. Section IV explains the datasets used in this study. Section V presents the numerical results, their comparison, and discussion. Section VI concludes this paper.

## II. Materials and Methods

### A. RSPRT With Relative Density Ratio Estimation (M1)

MODIS time-series data contain seasonal variations which need to be taken into account while designing any change detection method [17], [38]. Different types of functions have been used to model MODIS vegetation index time series, in land cover change detection framework, over the recent past [10]–[12], [14], [16]–[18]. Some recent studies have argued the usefulness of triply modulated cosine function and its time varying parameters, in land cover change detection framework [14], [16]–[18]. In order to get trend and seasonal variations separately, we model the vegetation index time series of a given MODIS pixel by a triply modulated cosine function as in [14], [16]–[18]

$$y_t = \mu_t + \alpha_t \sin(2\pi f t + \phi_t) + v_t \tag{1}$$

where $y_t$ and $v_t$ are the observation and noise value from an unknown distribution, at time $t = 1, 2, \ldots$. The above model is based on many unknown parameters, namely the frequency $f$, and the time-varying parameters mean $\mu_t$, amplitude $\alpha_t$, and phase $\phi_t$. The parameter $f$ is determined by the data being used for analysis. In our case, the MODIS 8-day 500-m time series has a cycles length of 1 year with 46 observations per year, hence $f = 1/46$. The values of $\mu_t$, $\alpha_t$, and $\phi_t$ can be estimated from the observations $y_t$ according to (1) using a nonlinear estimator. As proposed in [18], EKF can be used to derive the time-varying parameters of (1). In EKF formulation, the model given in (1) can be written as a pair of state and measurement equations as

$$\mathbf{x}_t = \mathbf{v}(\mathbf{x}_{t-1}) + \mathbf{w}_t \tag{2}$$

and

$$\mathbf{y}_t = \mathrm{h}(\mathbf{x}_t) + v_t \tag{3}$$

where $\mathbf{x}_t = [\mu_t, \alpha_t, \phi_t]^T$ is the state vector, $\mathbf{v}$ is the relationship between the previous state and the current state, $\mathbf{w}_t$ is $3 \times 1$ vector of process noise at time $t$, $v_t$ is the measurement noise at time $t$, $\mathbf{h}$ is the relationship between the current state $\mathbf{x}_t$, and the predicted measurement $\mathbf{y}_t$. The EKF predicts the state vector at time $t$ recursively [18], using the observations till time $t$.

Estimation of the state vector at every time point $t$ using EKF results in time series of the parameters. The next step is to compute the change metrics/test statistics to classify change or no-change events. As shown in [1], [14], and [15], the trend changes, e.g., changes due to beetle infestations create significant impact on $\mu_t$; hence, we calculate our test statistics from $\mu_t$. Many types of control charts exist in literature, e.g., Shewhart control charts [33], [36], [39], moving average control charts [33], [36], RSPRT/CUSUM control charts [33], [36], [37], generalized likelihood ratio (GLR) control charts [33], [36] etc., which can be applied in deriving the test statistics. However, RSPRT/CUSUM detect small changes quicker (takes lesser number of observations or data points after the change has occurred) than rest of the control charts [33], [36]. Since the type of change we are targeting here is of gradual nature, we use RSPRT to derive test statistic $S_t$ from $\mu_t$ time series as

$$S_t = \begin{cases} S_{t-1} + \ln\frac{p_{H_1}(\boldsymbol{\mu}_t)}{p_{H_0}(\boldsymbol{\mu}_t)}, & \text{if } S_{t-1} + \ln\frac{p_{H_1}(\boldsymbol{\mu}_t)}{p_{H_0}(\boldsymbol{\mu}_t)} > 0 \\ 0, & \text{if } S_{t-1} + \ln\frac{p_{H_1}(\boldsymbol{\mu}_t)}{p_{H_0}(\boldsymbol{\mu}_t)} \leq 0 \end{cases} \tag{4}$$

where $p_{\mathcal{H}_*}(\boldsymbol{\mu}_t)$ is the likelihood of vector random variable $\boldsymbol{\mu}_t$ at time $t$, under hypothesis $\mathcal{H}_*$, and $S_0 = 0$. The vector random variable $\boldsymbol{\mu}_t = [\mu_t, \mu_{t-1}, \ldots, \mu_{t-k+1}]$ in (4) is derived with a sliding window of length $k$ in order to capture the relationship of $\mu_t$ with its immediate past. The value of $k$ can be chosen by the user (normally $k \geq 10$, we used $k = 10$). The no-change and the alternate hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$, respectively, can be defined as

$$\begin{aligned} \mathcal{H}_0 &: \quad S_t \leq \lambda \\ \mathcal{H}_1 &: \quad S_t > \lambda \end{aligned} \tag{5}$$

where $\lambda$ is a carefully selected threshold. Equation (4) can be compacted as

$$S_t = (S_{t-1} + s_t)^+ \tag{6}$$

where $(\varrho)^+ = \sup(0, \varrho)$ for some value of $\varrho$, $s_t = \ln\frac{p_{\mathcal{H}_1}(\boldsymbol{\mu}_t)}{p_{\mathcal{H}_0}(\boldsymbol{\mu}_t)}$. The change alarm $a_t$ at time $t$ can be raised according to

$$a_t = \begin{cases} 1 & \text{if } S_t > \lambda, \\ 0 & \text{if } S_t \leq \lambda, \end{cases} \quad t \geq k \tag{7}$$

for a carefully selected threshold $\lambda$.

The likelihood ratio in RSPRT is often found either by assuming the individual density functions to be Gaussian [33] or by estimating the individual density functions using kernel density estimation (KDE) [13]. Both the methods can lead to suboptimal results because the real-world data rarely satisfy Gaussanity condition, and density estimation too is a difficult problem to solve [23], [24], [27], [40]. Estimating the density ratios directly, without estimating the individual distributions, is comparatively easier and achieves better performance [23], [24], [27], [40]. Although many algorithms have been used for direct density ratio estimation, e.g., KLIEP [23], [25], ULSIF [24], [26], [24], [27] suggested that RULSIF algorithm, which considers relative density ratios, achieves better estimation and nonparametric convergence. Therefore, we use relative density ratios, estimated directly from change and no-change training sets using RULSIF algorithm [27], in (4). Let $\mathbf{Y}_{tr}$ be the training set containing change and no-change ground truth examples. After deriving the parameters of (1) using EKF, the change and no-change training sets of the $\mu_t$ parameter denoted, respectively, by $\mathrm{Y_c} = \{\boldsymbol{\mu}_i\}_{i=1}^n$ and $\mathrm{Y_{nc}} = \{\boldsymbol{\mu}'_i\}_{i=1}^m$, can be formed by sliding a window of length $k$ over change and

no-change $\mu_t$ time series as $\boldsymbol{\mu}_t = [\mu_t, \mu_{t-1}, \ldots, \mu_{t-k+1}]$ starting at $t = k$, and putting the window values, at each time point, in the respective sets. According to the RULSIF formulation the relative density ratio can be given by [24] and [27]

$$r_\beta(\boldsymbol{\mu}) = \frac{p(\boldsymbol{\mu})}{\beta p(\boldsymbol{\mu}) + (1-\beta)p'(\boldsymbol{\mu})} = \frac{p(\boldsymbol{\mu})}{p'_\beta(\boldsymbol{\mu})} \tag{8}$$

where $p(\boldsymbol{\mu})$ represents the density of change samples, $p'(\boldsymbol{\mu})$ represents the density of no-change samples, $\boldsymbol{\mu}$ is an arbitrary data sample, and $0 \leq \beta < 1$. The $\beta$-relative density ratio $r_\beta(\boldsymbol{\mu})$ can be estimated by a kernel model as [24], [27]

$$r_\beta(\boldsymbol{\mu}) \approx g(\boldsymbol{\mu}; \boldsymbol{\theta}) = \sum_{i=1}^{n} \theta_i K(\boldsymbol{\mu}, \boldsymbol{\mu}_i) \tag{9}$$

$$K(\boldsymbol{\mu}, \boldsymbol{\mu}_i) = \exp\left(-\frac{\|\boldsymbol{\mu} - \boldsymbol{\mu}_i\|^2}{2\sigma^2}\right) \tag{10}$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_n]$ is the parameter vector, $\sigma$ (0) is the kernel width, and $n$ is the number of change examples in the training set. The appropriate value of $\sigma$ was selected as explained in [24].

Note that the complexity of problem increases with the increase in size of the training dataset because the number of kernels being used and hence the number of $\theta$ parameters which need to be estimated is equal to the number of samples in training set. This works well for small training sets, but it introduces memory and computation time issues in case of the large training sets. Therefore, we use a small but sufficient number of centers selected randomly from the training set, instead of using all the samples of the training set as centers. We adapt (9) as

$$r_\beta(\boldsymbol{\mu}) \approx g(\boldsymbol{\mu}; \boldsymbol{\theta}) = \sum_{i=1}^{d} \theta_i K(\boldsymbol{\mu}, \boldsymbol{\eta}_i) \tag{11}$$

where $\{\boldsymbol{\eta}_i\}_{i=1}^{d}$ is a set of $d$ number of centers chosen randomly from the training set. As formulated in RULSIF, the squared loss between the true and estimated relative density ratios $J(\boldsymbol{\mu})$ is given by [24] and [27]

$$\begin{aligned} J(\boldsymbol{\mu}) &= \frac{1}{2}\int p'_\beta(\boldsymbol{\mu})\left(r_\beta(\boldsymbol{\mu}) - g(\boldsymbol{\mu};\boldsymbol{\theta})\right)^2 \mathrm{d}\boldsymbol{\mu} \\ &= \frac{1}{2}\int p'_\beta(\boldsymbol{\mu})\left(r_\beta(\boldsymbol{\mu})\right)^2 \mathrm{d}\boldsymbol{\mu} \\ &\quad - \int p(\boldsymbol{\mu}) g(\boldsymbol{\mu};\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\mu} \\ &\quad + \frac{\beta}{2}\int p(\boldsymbol{\mu})\left(g(\boldsymbol{\mu};\boldsymbol{\theta})\right)^2 \mathrm{d}\boldsymbol{\mu} \\ &\quad + \frac{1-\beta}{2}\int p'(\boldsymbol{\mu})\left(g(\boldsymbol{\mu};\boldsymbol{\theta})\right)^2 \mathrm{d}\boldsymbol{\mu}. \end{aligned} \tag{12}$$

The parameter vector $\boldsymbol{\theta}$ can be estimated by minimizing $J(\boldsymbol{\mu})$. Ignoring the terms independent of $g(\boldsymbol{\mu};\boldsymbol{\theta})$ in (12), the following optimization problem is formulated according to RULSIF [27]

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^n}\left[\frac{1}{2}\boldsymbol{\theta}^T\widehat{H}\boldsymbol{\theta} - \widehat{h}^T\boldsymbol{\theta} + \frac{\gamma}{2}\boldsymbol{\theta}^T\boldsymbol{\theta}\right] \tag{13}$$

$$\implies \boldsymbol{\theta} = \left(\widehat{H} + \gamma I_n\right)^{-1}\widehat{h} \tag{14}$$

where $\widehat{H}$ is a $d \times d$ matrix, $I_d$ is an $d$ dimensional identity matrix, $\widehat{h}$ is a vector of length $d$, and $\gamma \geq 0$ is a regularization parameter. The $(l, l')$th element of $\widehat{H}$, for all $1 \leq l, l' \leq d$, is given by [27]

$$\begin{aligned} \widehat{H}_{l,l'} &= \frac{\beta}{n}\sum_{i=1}^{n} K(\boldsymbol{\mu}_i, \boldsymbol{\eta}_l) K(\boldsymbol{\mu}_i, \boldsymbol{\eta}_{l'}) \\ &\quad + \frac{1-\beta}{m}\sum_{j=1}^{m} K(\boldsymbol{\mu}'_j, \boldsymbol{\eta}_l) K(\boldsymbol{\mu}'_j, \boldsymbol{\eta}_{l'}) \end{aligned} \tag{15}$$

and $l$th element of $\widehat{h}$ can be given by [27]

$$\widehat{h}_l = \frac{1}{n}\sum_{i=1}^{n} K(\boldsymbol{\mu}_i, \boldsymbol{\eta}_l). \tag{16}$$

Once the parameter vector $\boldsymbol{\theta}$ has been estimated, it is used in (11) to estimate the relative density ratio of any $\boldsymbol{\mu}_t$ which is then used in (4) to calculate the test statistics. First, we find the relative density ratio sequences for all the examples in the training set. The threshold $\lambda$ is then tuned using this training set. Then, in the similar way relative density ratio is found at any time $t$ in test time series as well, and change alarm can be raised according to (7). Both the training and testing phases of the proposed framework have been summarized in Algorithms 1 and 2, respectively. It is worth noting here that the authors of [13] and [41] mentioned that both independent and identically distributed (i.i.d.) assumptions were not met in their formulation of CUSUM. After removing the seasonality, the unchanged $\mu_t$ parameter time series has slightly reduced correlation but not enough to be considered as negligible. However, our formulation considers all the no-change samples as coming from a single distribution, unlike CUSUM formulation in [13], [41]. Therefore, the no-change samples can be considered as identically distributed, and the change is detected when this assumption is violated, i.e., when a sample from a significantly different distribution is encountered.

**Algorithm 1.** Training($\mathbf{Y}_{tr}$,$k$, $\beta$, $\sigma$, $\gamma$)

**Given** the training set $\mathbf{Y}_{tr}$, $k$, $\beta$, $\sigma$, and $\gamma$

1) Derive the parameters $\mathbf{x}_t = [\mu_t, \alpha_t, \phi_t]$ for all training time series using EKF.
2) Make separate sets for change and no-change samples, $\mathrm{Y}_c = \{\boldsymbol{\mu}_i\}_{i=1}^{n}$ and $\mathrm{Y}_{nc} = \{\boldsymbol{\mu}'_i\}_{i=1}^{m}$, respectively.
3) Chose $d$ samples from $\mathrm{Y}_c$ randomly as kernel centers ($\boldsymbol{\eta} = \{\boldsymbol{\eta}_i\}_{i=1}^{d}$).
4) Use $\boldsymbol{\eta}, \mathrm{Y}_c$, and $\mathrm{Y}_{nc}$ in (13) to (16) to estimate the parameter vector $\boldsymbol{\theta}$ of the (11).
5) At every time point $t$ of the training time series estimate $r_\beta(\boldsymbol{\mu}_t)$ using (11).
6) Use the estimated $r_\beta(\boldsymbol{\mu}_t)$ in place of $\frac{p_{\mathcal{H}_1}(\boldsymbol{\mu}_t)}{p_{\mathcal{H}_0}(\boldsymbol{\mu}_t)}$ in (4) to calculate RSPRT statistic $S_t$.
7) Using RSPRT statistics of the whole training set, tune an optimal threshold ($\lambda$) that minimizes false negatives (FN), false positives (FP), and mean detection delay (MD).

OUTPUT ($\boldsymbol{\theta}$, $\lambda$, $\boldsymbol{\eta}$)

**Algorithm 2.** Test($y_t$, $k$, $\boldsymbol{\eta}$,$\mathrm{Y}_c$ and $\sigma$)

**Given** all the observations till current time $t$ ($\mathbf{y}_t$), $k$, $\boldsymbol{\eta}$, $\mathrm{Y}_c$, and $\sigma$

1) Derive the parameters $\mathbf{x}_t = [\mu_t, \alpha_t, \phi_t]$ of (1) using EKF.
2) Derive a test sample $\boldsymbol{\mu}_t = [\mu_t, \mu_{t-1}, \ldots, \mu_{t-k+1}]^T$.
3) Estimate $r_\beta(\boldsymbol{\mu}_t)$ using (11).
4) Use the estimated $r_\beta(\boldsymbol{\mu}_t)$ in place of $\frac{p_{\mathcal{H}_1}(\boldsymbol{\mu}_t)}{p_{\mathcal{H}_0}(\boldsymbol{\mu}_t)}$ in (4) to calculate RSPRT statistic $S_t$.
5) Use (7) to evaluate the change alarm $a_t$.

OUTPUT ($a_t$)

### *B. Automatic Threshold Tuning*

Tuning a threshold manually on training dataset, while taking care of three performance indices, i.e., $FN$, $FP$, and $MD$, is a challenging task. Here, we formulate a simple, yet effective constrained optimization problem, which will yield a tuned threshold. Since all the three performance indicators are desired to be as low as possible, an over all cost function-based including the effect of all of them can be designed, which can then be minimized. A simple choice can be Euclidean norm written as

$$\mathcal{L}_q = \sqrt{(FP_q)^2 + (FN_q)^2 + (\psi \times MD_q)^2} \qquad (17)$$

where $\mathcal{L}_q$ is the cost at the $q$th iteration of the optimization algorithm and $\psi$ is the weight that increases or decreases the dependence of the cost function/optimization on $MD$. A properly selected $\psi$ also caters for the scale difference between $MD$ and rest of the two indicators. Its value can be selected against a desired accuracy in the first run of the cross-validation and kept the same throughout the rest of the experiment. Note that $FP$, $FN$, and $MD$ are derived from (7) which means that they are dependent on $\lambda$ and hence the cost function as well. So, ideally, minimization of the cost function in (17) subject to $\lambda > 0$ should yield optimal value of $\lambda$ for a specific value of $\psi$. However, at some instances, the cost function may remain (flat) unchanged with the change in the value of $\lambda$, as shown in Fig. 1 (top). This is undesirable because the optimization algorithm may get stuck in such "flat" regions and stop prematurely. Mathematically, the flat regions in Fig. 1 (top) violate

$$P\left\{\left|\frac{d\mathcal{L}_q}{dq}\right| > 0 \mid \mathbf{E}\left[\frac{d\mathcal{L}_q}{dq}\right] < 0\right\} = 1 \qquad (18)$$

for all $q < N$, where $N$ is the iteration number at which the optimization algorithm converges and $\mathcal{L}_N < \mathcal{L}_{q \neq N}$. The operators $P\{\bullet\}$ and $\mathbf{E}\{\star\}$ represent probability and expectation, respectively. In order to tackle this issue we slightly modify (17) and introduce stochasticity in it as

$$\mathcal{L}'_q = \sqrt{(FP_q)^2 + (FN_q)^2 + (\psi \times MD_q)^2} + \varepsilon_q \qquad (19)$$

where $\mathcal{L}'_q$ is the value of stochastic cost and $\varepsilon_q$ is a small random number drawn from uniform or Gaussian distribution, in

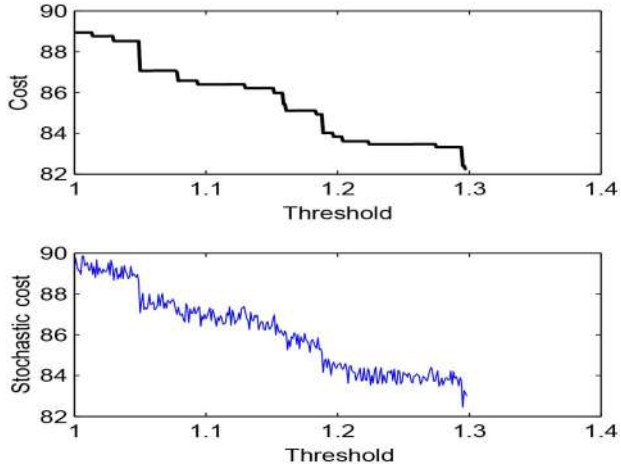


Fig. 1. Threshold ($\lambda$) versus cost (top). Threshold ($\lambda$) versus stochastic cost (bottom).

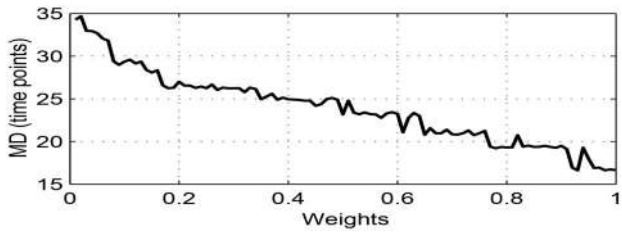


Fig. 2. Weight ($\psi$) versus $MD$ plot. Increasing weights ($\psi$) decreases the acceptable $MD$ values. The unit of MD is time points = number of time points or number of observations.

$q$th iteration. The effect of randomness/stochasticity has been shown in Fig. 1 (bottom). It can be seen in Fig. 1 (bottom) that (19) satisfies (18). Note that the condition in (18) is not strict. So, there is still a possibility that the optimization algorithm may get trapped in local minimum and stop prematurely, without converging to an optimal value of $\lambda$. Therefore, the minimization must be carried out several times (e.g., 10–20 times), each time with different initial value of $\lambda$, and the one with the lowest value of the cost function after convergence should be selected. Our experiments with different optimization algorithms suggest that genetic algorithm (GA) is able to optimize (17) successfully.

The weight $\psi$ varies the importance of $MD$ in the cost function, i.e., increasing $\psi$ will cause minimization to occur at lower values of $MD$ and vice versa. This relationship can be seen in Fig. 2 which shows $\psi$ versus $MD$ plot. Mathematically, it can be written as

$$\mathbf{E}\left[\frac{dMD}{d\psi}\right] < 0. \qquad (20)$$

An alternative for the above threshold selection strategy can be based on kappa-statistic ($\kappa$) [35], [42] and $MD$. Use of $\kappa$-statistic in remote sensing to measure homogeneity is somewhat controversial and there has been some criticism by

$$\kappa = \frac{N \times (TP' + TN') - \{(TP' + FP') \times (TP' + FN') + (TN' + FP') \times (TN' + FN')\}}{N^2 - \{(TP' + FP') \times (TP' + FN') + (TN' + FP') \times (TN' + FN')\}} \qquad (21)$$

some authors on the accuracy assessment based on $\kappa$-statistic, because its value depends strongly on the marginal distributions [43]–[45]. However, it is still the most widely used statistic, hence the usage here. It can be calculated as [(21), shown at the bottom of the previous page] where $TP'$, $TN'$, $FP'$, and $FN'$ are the true values (not percentages) of true positives, true negatives, FP, and FN, respectively, and $N = TN' + FN' + TP' + FP'$. The optimal threshold, corresponding to acceptable values of $MD$ or $\kappa$-coefficient, can be selected from "$\kappa$-coefficient versus $MD$" plots. This can also be achieved by minimizing a cost function based on $\kappa$ and $MD$ using GA, subject to desired constraints on $MD$ and/or $\kappa$, which can be easily programmed and integrated in cross-validation experiments. We used this strategy in our cross-validation experiments (see Section V-D). The advantage of $\kappa$-statistic-based threshold selection strategy is that it incorporates all the performance indices into a single coefficient and the threshold selection becomes more convenient as it can be selected from a plot of two variables. However, if there are some special constraints on $TP$ and $TN$ (or $FP$ and $FN$) which need to be followed in a certain application, then the former threshold selection strategy is more convenient.

## III. Existing Methods

### A. Original CUSUM With KDE (M2)

A supervised method, implementing CUSUM, was proposed in [13] for land cover change detection in MODIS NDVI time-series data. In this method, CUSUM statistics are derived from raw MODIS NDVI time series. The individual density functions required for calculating likelihood ratios are derived for every time point in the year/cycle using KDE. The MODIS product used in this study has the time resolution of 46 images/ year. Therefore, using this method, estimation of total of 92 density functions (46 density functions separately for both change and no-change hypotheses) is required. Once the density functions are estimated, the likelihood ratio (density ratio) of an observation at any time point in the cycle is calculated using the trained density functions of change and no-change for that particular time of the cycle/year. The likelihood ratio is then used to calculate the CUSUM statistic sequentially, which is then compared with a tuned threshold to detect any change event. Since this study uses MODIS NDVI data and the method is based on CUSUM statistics with estimated individual densities, comparison with this method will give us a good insight into advantages of model fitting and using estimated/trained relative density ratios rather than estimated/trained individual densities.

### B. Near Real-Time Disturbance Detection (M3)

Another method that was published recently in [10] is the "near real-time disturbance detection in MODIS data." In this method, MODIS NDVI time series is modeled using a function with constant, ramp, sine, and cosine terms. First, the function is fitted to the reference (no-change) period using nonlinear least squares fitting and its unknown parameters are derived. Then, using these parameters, the future observations are predicted using the model. The difference between the predicted and observed values gives noise time series. The MOSUM (MOving SUM) statistics [46], [47] are then derived from the noise time series and compared with a threshold, tuned according to functional central limit theorem [48], in order to detect any change events. Since this method uses noise time series to derive test statistics and also Gaussanity assumption which is implicit in central limit theorem, comparison with it will give us a good insight into benefits of using parameter time series and also relative density ratio estimation over assuming individual densities to be Gaussian.

### C. Near Real-Time Detection of Beetle Infestation (M4)

One of the most recent studies that addresses near real-time detection of land cover changes, specifically beetle infestations in pine forests, using MODIS NDVI data was published in [14]. In this method, nonlinear least squares approach is used to fit a model to the NDVI time series and derive its time-varying parameters. Based on the fact that beetle infestation affects the trend of the signal significantly, the trend component of the model is used to derive the test statistics. It is assumed that the underlying densities of the change and no-change parts of the trend component are nearly Gaussian with difference in their means. Based on this assumption, the log-likelihood ratio of the value of the trend component, at any particular time point, is calculated. This log-likelihood ratio is then compared to a tuned threshold to declare a change or no-change event. The threshold is tuned by finding a good tradeoff between the likelihood ratios of change and no-change training sets.

## IV. Datasets

### A. Simulated Data

One main problem that is often encountered in case of near real-time change detection is that the ground-truth data with accurate labels/time point of change events is hard to find [11]. Most often, partial information is known, e.g., changed and unchanged pixels are known, but the exact time points at which the changes occurred in the respective time series are unknown, hence making the performance evaluation difficult and compromised. Many studies have used and highlighted the importance of simulated data, in which, changes are introduced at desired time points [10]–[12], [14], [16], [49]. Such data can be helpful in evaluating performance, sensitivity, and robustness of the method to different magnitudes of noise. We also generated a simulated dataset following a similar procedure as used in [10], [11], [12], [14], and [16].

First, the deterministic part or seasonal cycles were generated using asymmetric Gaussian function as proposed in [12] and [14]

$$g(l) \equiv g(l; a, b, \rho_1, \rho_2) = a \times \begin{cases} \exp\left[-\frac{(l-b)^2}{\rho_1}\right], & \text{if } l > b \\ \exp\left[-\frac{(b-l)^2}{\rho_2}\right], & \text{if } l < b \end{cases} \tag{22}$$

where $\rho_1$ and $\rho_2$ control the width of the left and right hand sides, whereas $a$ and $b$ are the amplitude and the position of the

maximum or minimum with respect to the time $l$, respectively. We used $a = 0.7$, $b = 23 + \lfloor l/46 \rfloor \times 46$, and $\rho_1 = \rho_2 = 100$ in (22). The simulated time series were generated by

$$S(l) = g(l) + \Phi(l) + \vartheta_l \tag{23}$$

where $\vartheta_l$ is a noise value at $l$, drawn from Gaussian distribution with zero mean. The $\Phi(l)$ in (23) is the simulated gradual change introduced in the time series, and can be given by

$$\Phi(l) = \begin{cases} \left[(1)^{\lfloor l/\varrho \rfloor} - (0)^{\lfloor l/\varrho \rfloor}\right] \times \varsigma \times (l - \varrho), & \text{if } l \leq \xi \\ \Phi(\xi), & \text{if } l < \xi \end{cases} \tag{24}$$

where $\varsigma$, $\varrho$, and $\xi$ are the slope, start point, and end point of the introduced change, respectively. Note that simulating remotely sensed data with vegetation phenology, inter-annual variability, disturbance events, and signal contamination is challenging [11]. Therefore, testing the method on a variety of datasets is necessary.

### B. Synthetic Data

Simulated dataset is far from real-world data and the factors involved in it. To get as much close as possible to the real-world data, yet knowing the exact time points of the change events, another type of data, namely, synthetic dataset, has also been used [13], [17], [18], [50], [51]. This data can be created from sure change and no-change parts of the real-world time series. First, all the time series are standardized according to the range of the no-change part and then different no-change parts are concatenated with different change parts randomly, creating a large number of time series. So, all the time series have natural factors involved in them as well as the exact time points of change events are also known.

### C. Real-World Beetle Infestation Data

We used yearly survey maps and shape files maintained by the U.S. and British Columbia forest services [52]–[54] to identify the areas with beetle infestations in the pine forests of Colorado, Utah (United States), and British Columbia (Canada). The regions with no beetle infestation history till the end of 2005 were selected. These regions were then marked on the Google Earth and their geographical coordinates were recorded. An online tool (MODLAND Tile Calculator) [55] was then used to identify the corresponding MODIS tile using the geographical coordinates. Once the MODIS tile was known, one MODIS image of 500-m spatial resolution was fed into a software, namely ENVI (version 5). Using the geographical coordinates of the marked regions, their pixel coordinates, in any MODIS image of 500-m spatial resolution, were found with the help of ENVI. After recording all this information, the MODIS product MCD43A4.005 was downloaded for the desired tiles starting from January 2001 to December 2011, and the time series of the selected pixels were extracted. All the change and no-change examples from both the regions were combined and two (change and no-change) datasets were prepared.

The MODIS product MCD43A4.005 is available since 2000. It provides 500-m 8-day composite reflectance data which are bidirectional reflectance distribution function (BRDF)-adjusted for Nadir reflectance, atmospherically corrected and cloud free. The data acquired for the year 2000 had a lot of missing values; hence, it was discarded and the time-series data were acquired from January 2001 onward. Although we did not encounter any missing values in our analysis, however, rare missing values can be replaced with interpolated values.

## V. Results, Comparison, and Discussion

### A. Results for Simulated Data

We generated 500 change and 500 no-change examples according to the methodology used in [10]–[12], [14], and [16] summarized in Section IV-A. The seasonal cycles were generated using asymmetric Gaussian function, the change was introduced by adding a ramp of slope 0.0025 [$\varsigma = 0.0025$ in (24)] to the signals at known positions in order to replicate a gradual change, and the noise introduced into the signal was drawn randomly from the noise distribution with standard deviation of 0.08. Randomly selected 50% samples of the dataset was taken as the training set and the rest 50% was taken as the test set. The purpose of this dataset was to analyze the performance of the proposed framework (M1) on a dataset with known change points and to compare it with the performances of the existing methods. Moreover, it can also be used to analyze the robustness of the proposed method M1 against different magnitudes (standard deviations) of noise.

All the four methods, M1–M4, were implemented keeping the training and test sets exactly the same to ensure fair comparison. A wide range of threshold values were used for each method to exploit its performance range and capabilities. The results of all the methods, M1–M4, have been summarized in Figs. 3 and 4 and Table I. Note that all the four methods have different ranges of threshold values, but here we have scaled all of them to a single range of 0–80, for the sake of simplicity in comparison. The absolute values of the thresholds are not important here because we only want to graphically present the best possible performances by each of the methods considered here. Fig. 3 (top) presents the "threshold versus $\kappa$-coefficient" plots of all the methods. We note that all the methods can achieve accuracies close to $\kappa = 1$. The value of $\kappa$-coefficient increases with the increase in the threshold value, but as a consequence, the mean detection delay also increases as shown in Fig. 3 (bottom). Fig. 4 summarizes the plots of Fig. 3 and gives a more obvious comparison by plotting MD against the corresponding $\kappa$-coefficients, for each method considered here. Fig. 4 basically shows different tradeoffs between kappa (accuracy) and MD for each method, which is analogous to the ROC curve that plots tradeoffs between TP and FP when only accuracies (without MD) are considered. Focusing on the significant region of this plot, i.e., after $\kappa = 0.6$, we observe that although the difference between the plots is not very large, M1 and M4 perform slightly better than the rest of the two methods. This fact is also obvious from the comparison shown in Table I for the same accuracy of 99%.

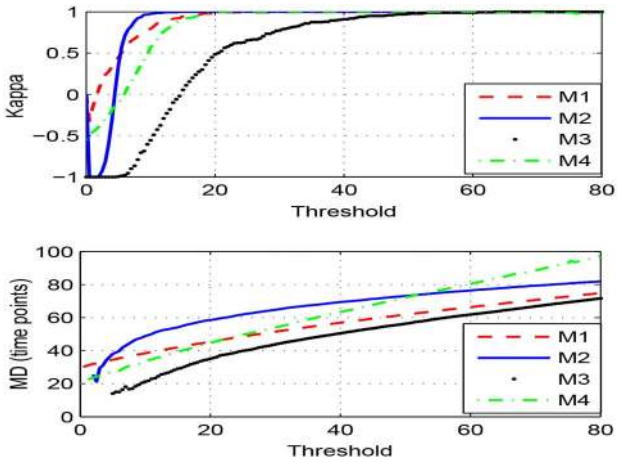


Fig. 3. Comparison between performances of proposed (M1), original CUSUM (M2), near real-time disturbance detection (M3), and near real-time beetle infestation detection (M4) methods, on simulated data. Threshold ($\lambda$) versus $\kappa$-coefficient (top). Threshold ($\lambda$) versus MD (bottom). The unit of MD is time points = number of time points or number of observations.

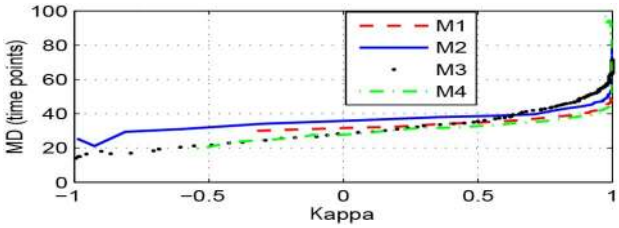


Fig. 4. Comparison of "kappa-coefficient versus MD" performances of proposed (M1), original CUSUM (M2), near real-time disturbance detection (M3), and near real-time beetle infestation detection (M4) methods on simulated data. The unit of MD is time points = number of time points or number of observations.

TABLE I
COMPARISON BETWEEN NEAR REAL-TIME PERFORMANCES OF OUR PROPOSED METHOD (M1) AND THE EXISTING THREE METHODS, ON SIMULATED DATA, AT ACCEPTABLE TRUE POSITIVES, TRUE NEGATIVES, AND ACCURACY

| Method | TP | TN | Acc. | MD (tp) | Threshold ($\lambda$) |
|---|---|---|---|---|---|
| M1 | 100 | 98 | 99 | 45 | 18.50 |
| M2 | 98.50 | 99.50 | 99 | 49 | 10.50 |
| M3 | 100 | 98 | 99 | 56 | 05.05 |
| M4 | 100 | 98 | 99 | 44 | 03.80 |

TP, true positive; TN, true negative, Acc., overall accuracy; MD, mean detection delay; and $\lambda$, threshold value. M1, proposed framework; M2, original CUSUM method; M3, near real-time disturbance detection method (M3), and M4, near real-time beetle infestation detection method. The units of TP, TN, Acc. are "%" and that of MD is tp, number of time points or observations.

We note that the results achieved here were according to our expectations because the simulated data lack the effects of the complex natural phenomenon which are present in the real-world NDVI data. Moreover, the noise in simulated data is Gaussian that satisfies the Gaussanity assumptions in M2–M4, and the points of actual changes in the training data are known exactly, hence no mislabeling that can affect the training adversely. Nevertheless, these results illustrate the correctness

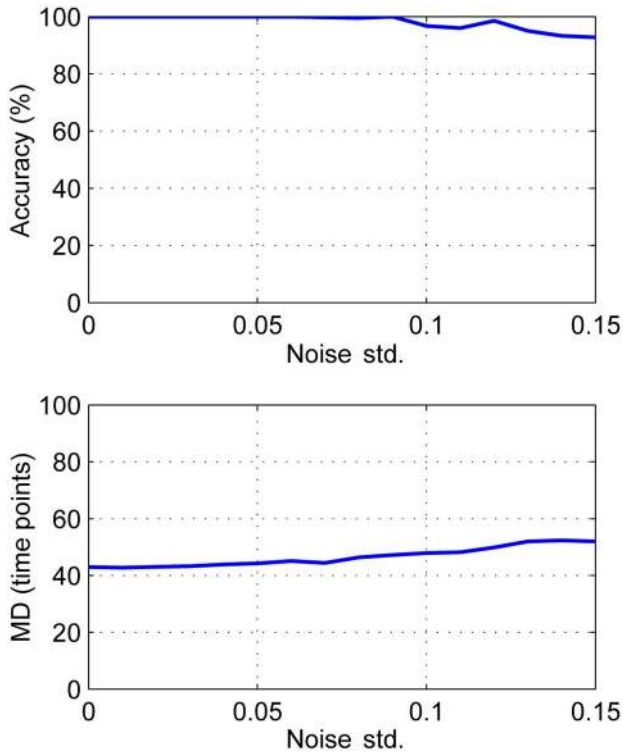


Fig. 5. Robustness of the proposed method against different magnitudes of noise in the simulated dataset. Noise standard deviation (Noise Std.) versus overall accuracy (top). Noise standard deviation (Noise Std.) versus detection delay (bottom). The unit of MD is time points = number of time points or number of observations.

of the approaches in that their tendency is to detect the real changes and avoid no-change events.

The performance of M1 was also checked on simulated datasets with different magnitudes (standard deviations) of noise. The results are summarized in Fig. 5. Fig. 5 (top) shows that the accuracy drops very slightly with the increase in the standard deviation of the signal noise, and remains above 90% even at standard deviation as high as 0.15. Fig. 5 (bottom) shows a very slight increase in the detection delay, from 42 at noise std. = 0 to 53 at noise std. = 0.15. These results show the robustness of the proposed framework M1 to different magnitudes of signal noise.

### B. Results for Synthetic Data

We prepared 1000 change and 1000 no-change examples following the process explained in Section IV-B, and also used in [13], [17], [18], [50], and [51]. A randomly selected set of 50% of the samples was taken as training set and the rest 50% as test set. The results have been summarized in Figs. 6 and 7 and Table II. Fig. 6 (top) shows that the value of $\kappa$ increases with the increase in the threshold, except for M4 that decreased after reaching its peak. The increase in threshold also causes increase in the detection delays, as shown in Fig. 6 (bottom), because the test statistic has to attain bigger values in order to raise the change alarm. Fig. 7 summarizes the plots of Fig. 6 by plotting MD against the corresponding $\kappa$-coefficients, for each method considered here. The plots in Fig. 7 are analogous to

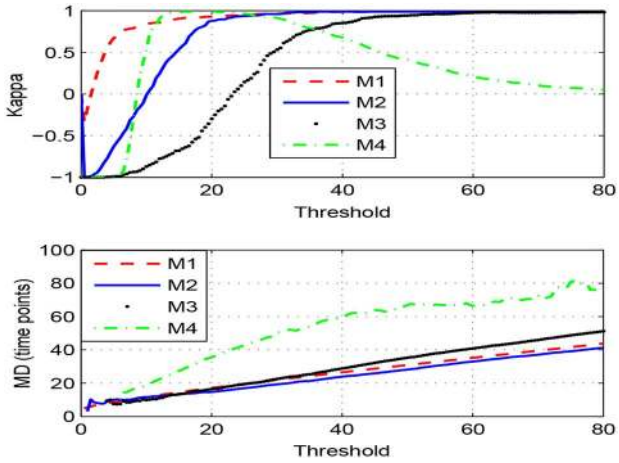


Fig. 6. Comparison between performances of proposed (M1), original CUSUM (M2), near real-time disturbance detection (M3) and near real-time beetle infestation detection (M4) methods, on synthetic data. Threshold ($\lambda$) versus $\kappa$-coefficient (top). Threshold ($\lambda$) versus MD (bottom). The unit of MD is time points = number of time points or number of observations.

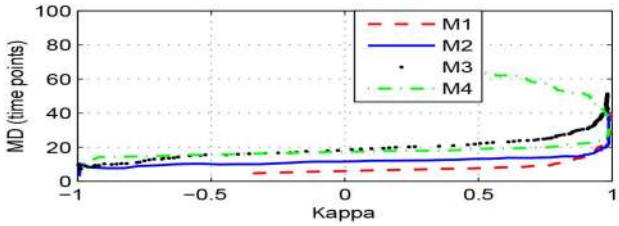


Fig. 7. Comparison of "kappa-coefficient versus MD" performances of proposed (M1), original CUSUM (M2), near real-time disturbance detection (M3) and near real-time beetle infestation detection (M4) methods on synthetic data. The unit of MD is time points = number of time points or number of observations.

TABLE II
COMPARISON BETWEEN NEAR REAL-TIME PERFORMANCES OF OUR PROPOSED METHOD (M1) AND THE EXISTING THREE METHODS, ON SYNTHETIC DATA, AT ACCEPTABLE TRUE POSITIVES, TRUE NEGATIVES, AND ACCURACY

| Method | TP | TN | Acc. | MD | Threshold ($\lambda$) |
|---|---|---|---|---|---|
| M1 | 100 | 94 | 97 | 17 | 20.50 |
| M2 | 97 | 97 | 97 | 17 | 25 |
| M3 | 100 | 94 | 97 | 33 | 04.75 |
| M4 | 100 | 94 | 97 | 23 | 02.50 |

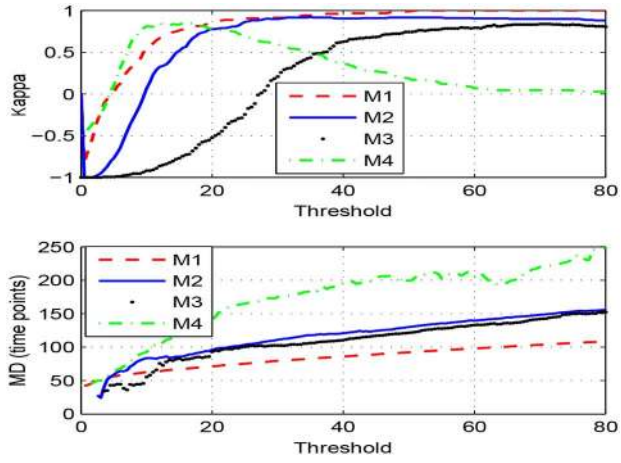


Fig. 8. Comparison between performances of proposed (M1), original CUSUM (M2), near real-time disturbance detection (M3), and near real-time beetle infestation detection (M4) methods, on real-world data. Threshold ($\lambda$) versus $\kappa$-coefficient (top). Threshold ($\lambda$) versus MD (bottom). The unit of MD is time points = number of time points or number of observations.

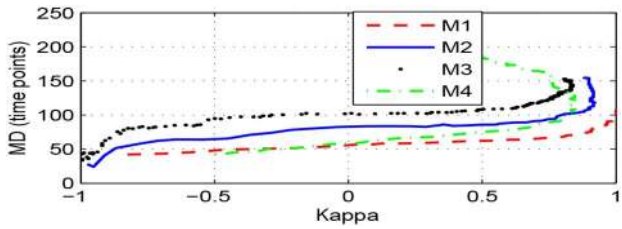


Fig. 9. Comparison of "kappa-coefficient versus MD" performances of proposed (M1), original CUSUM (M2), near real-time disturbance detection (M3), and near real-time beetle infestation detection (M4) methods on real-world data. The unit of MD is time points = number of time points or number of observations.

TABLE III
COMPARISON BETWEEN NEAR REAL-TIME PERFORMANCES OF OUR PROPOSED METHOD (M1) AND THE EXISTING THREE METHODS, ON NEAR REAL-TIME NDVI DATA, AT ACCEPTABLE TRUE POSITIVES, TRUE NEGATIVES, AND ACCURACY

| Method | TP | TN | Acc. | MD (tp) | Threshold ($\lambda$) |
|---|---|---|---|---|---|
| M1 | 100 | 90.6 | 95.3 | 79 | 26.1 |
| M2 | 84 | 96 | 90 | 99 | 22 |
| M3 | 90 | 90 | 90 | 131 | 5.8 |
| M4 | 98 | 84 | 91 | 92 | 02 |

ROC curves and give a more obvious comparison. Here, the difference between the performances is slightly more obvious than in case of simulated data because the data values and noise are real, only the changes introduced are synthetic, i.e., due to concatenation of known no-change and change parts. Considering the significant part of the curves, i.e., the region after $\kappa \approx 0.6$ on the horizontal axis, our proposed method M1 performs better than the rest until the point around $\kappa \approx 0.90$, where M1 and M2 become similar in performance. Table II also highlights this where detection delays are compared for all the four methods against the same accuracy of 97%. The M1 and M2 have similar results, better than M3 and M4. Furthermore, M4 performs better than M3. The reader should not be confused by the M4 curve hooking back in Fig. 7. This behavior is quite possible because MD is not a function of the $\kappa$-coefficient. The $\kappa$-coefficient is calculated from TP, TN, FP, and FN, which depend on the threshold value. Two different thresholds can yield exactly the same $\kappa$-coefficient with different values of MD. The hooked curve shows exactly the same behavior, i.e., same $\kappa$-coefficients with different values of MD.

The noise is far from Gaussian in this case, but still the change points in the training data are known exactly, hence no mislabeling. Therefore, M2 still trains very well, hence small difference between the results of M1 and M2. The fact that M2 performs better than M3 and M4 can be attributed to the difference between the types of test statistics being used in these methods. The M2 uses CUSUM statistics which is more

TABLE IV
RESULTS OF 50% CROSS-VALIDATIONS (10 RUNS) OF M1 ON ALL THE THREE DATASETS. IN EVERY RUN RANDOMLY SELECTED 50% OF THE DATA SAMPLES OF A PARTICULAR DATASET IS TAKEN AS TRAINING SET, AND THE REST AS TEST SET

| Dataset | Data | Mean (10 runs) | | | | | | Standard Deviation (10 runs) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | TP | TN | Acc. | $\kappa$ | MD (tp) | $\lambda$ | TP | TN | Acc. | $\kappa$ | MD (tp) | $\lambda$ |
| Sim. | Training | 100 | 95.75 | 97.9 | 0.957 | 45.7 | 16.67 | 0 | 1.09 | 0.54 | 0.011 | 0.28 | 0.52 |
| | Test | 100 | 96.00 | 98.00 | 0.960 | 45.8 | | 0 | 1.26 | 0.63 | 0.013 | 0.63 | |
| Syn. | Training | 100 | 95.5 | 97.75 | 0.955 | 21.8 | 29.83 | 0 | 1.47 | 0.74 | 0.015 | 0.25 | 0.77 |
| | Test | 100 | 95.2 | 97.30 | 0.952 | 22.7 | | 0 | 1.34 | 0.68 | 0.013 | 0.51 | |
| R.W. | Training | 100 | 89.50 | 94.75 | 0.895 | 80.5 | 27.42 | 0 | 2.4 | 1.2 | 0.024 | 0.47 | 2.1 |
| | Test | 100 | 88.00 | 94.00 | 0.880 | 79.1 | | 0 | 2.5 | 1.25 | 0.025 | 0.7 | |

TP, true positive; TN, true negative; Acc., overall accuracy; $\kappa$, kappa-coefficient, MD, mean detection delay; $\lambda$, threshold value; Sim. = simulated dataset; Syn., synthetic dataset; R.W., real-world NDVI beetle infestation data. The units of TP, TN, and Acc. are "%", and that of MD is tp = number of time points or observations.

robust to nonGaussanity and detects small/gradual changes earlier than the statistics based on simple likelihood ratios [36] as used in M3 and M4. The M4 performing better than M3 confirms the findings of [14].

### C. Results for Real-World MODIS NDVI (Beetle Infestation) Data

The simulated and synthetic datasets provided important insights, but these datasets do not include all the complexities that are encountered in the real-world data, e.g., atmospheric variations, light variations, and lack of information about the exact time points of the changes. Therefore, analysis on the real-world data is necessary. We tested all the methods on 355 change and 355 no-change examples of beetle infestation data, collected as explained in Section IV-C. The results have been summarized in Figs. 8 and 9 and Table III. As explained before, all the four methods have different ranges of threshold values, but we have scaled all of them to a single range of 0–80, for the sake of simplicity in comparison. The absolute values of the thresholds are not important here because we only want to graphically present the best possible performances by each of the methods considered here. Fig. 8 shows similar trends for each method as in the case of synthetic data since the two datasets are close in nature to each other. Fig. 8 (top) presents the "threshold versus $\kappa$-coefficient" plots of all the methods. The $\kappa$ values of M2, M3, and M4 drop after reaching their peaks, whereas the $\kappa$ value of M1 remains constant. Fig. 8 (bottom) presents the "threshold versus MD" plots of all the methods. The MD generally increases with the increase in threshold value, for all the methods. Fig. 9 summarizes the plots of Fig. 8 by plotting $\kappa$-coefficient against the corresponding accuracies, for each method considered here. The plots in Fig. 9 are analogous to ROC curves and give a clearer comparison. It can be seen in Fig. 9 that our proposed method M1 performs better than the other three methods by a significant margin. For the similar values of $\kappa$-coefficient, M1 incurs much lower detection delay than the rest of the three methods. Furthermore, M1 can still get close to $\kappa = 1$, unlike rest of the methods which peaked at significantly lower $\kappa$ values than M1. Table III compares the performances of all the methods at acceptable accuracies (95.3% for M1, 90%, 90%, and 91% for M2, M3, and M4, respectively). It can be noticed that M1 incurs much lower detection delay at a higher accuracy/kappa than rest of the three methods.

The reason behind such a significant difference between the performance of M1 and the rest of the methods can be attributed to three facts: 1) the test statistics in M1 derived from the parameter time series which has been shown to be better than the statistics derived from the raw NDVI time series [14]; 2) RSPRT statistics are used in M1, which are more robust to nonGaussanity and detects small changes faster than the simple statistics based on likelihood ratios [36]; and 3) the likelihood ratios used in deriving RSPRT statistics were estimated directly using RULSIF algorithm [27], which performs better than the likelihood ratios derived from individual density functions which are based on Gaussanity assumption or estimated individually using Gaussian kernels. Furthermore, it is very difficult to obtain the exact ground reference data for long time series [56]. The forestry departments, which are monitoring those forests, also confirmed that the survey maps were manual and subjected to errors; hence, the real-world data did not have exact information about the change points in each time series. Therefore, we took $t = 230$ as common reference point for all the methods, i.e., the last point for known no-change part in every time series, from which MDs were calculated. However, this does not mean that the changes were known to have occurred at that point. Majority of the time series changed at later unknown points, which implies that the training data had mislabeling at some points after $t = 230$. This suggests that M1 is more robust to mislabeling in the training data as compared to the other three methods.

### D. Cross-Validation and Automatic Threshold Tuning Results

We performed cross-validation experiments of the proposed method M1 on all the three datasets. The cross-validation consisted of 10 runs, where in each run 50% of the data samples selected randomly were used as training and the rest 50% as test sets. The results have been summarized in Table IV. The table consists of two vertical halves and three horizontal parts. The left vertical half summarizes the mean of performance indices for the training and test sets of all the datasets, whereas the right vertical half summarizes the standard deviations of the 10 runs. Each of the three horizontal parts of the table summarizes the performance indicators for one of the three datasets. The

value of $\psi$ was selected, in the first run of the cross-validation experiment on simulated data, against 98% accuracy and kept fixed for rest of the experiments. The mean performance indices of the training set and test set are very close to each other, in case of all the three datasets. This suggests that the thresholds selected from the training datasets are robust and perform equally well on the unseen test data.

The threshold selected in each run of this cross-validation experiment was selected automatically by our proposed threshold tuning technique (based on $\kappa$-coefficient and $MD$) as explained in Section II-B. Apart from tuning, another advantage of this technique, especially in cross-validation experiment, is that it can be incorporated in the code and a complete set of cross-validation results can be generated in a single go, without stopping and selecting the right threshold in each cross-validation run. The table shows that the thresholds selected by this technique on the training datasets perform nearly similar to the test datasets. This shows the effectiveness of the proposed threshold tuning technique. It is worth noting that the range of the suitable threshold values may change from one dataset to another, hence needs tuning on all the datasets separately. However, the proposed threshold tuning technique avoids this problem since the value of $\psi$ selected for one dataset holds good for rest of the datasets as well.

## VI. Conclusion

In this paper, we proposed a supervised framework for near real-time land cover change detection that uses EKF to fit a triply modulated cosine function to a MODIS NDVI time series, extracts its time varying parameters, and derives the RSPRT test statistics from the trend parameter. Instead of using traditional likelihood ratios, we exploited the usefulness of relative density ratios estimated directly using RULSIF algorithm as proposed in [27], in deriving the RSPRT statistics. Our framework slightly reduces the correlation in the parameter time series, and unlike CUSUM formulation in [13] and [41], deals with the no-change samples as identically distributed, which is an important assumption of CUSUM. We tested the framework on three different datasets, against different noise level, and also performed cross-validation. Furthermore, we compared its performance with three recently published near real-time change detection methods in remote-sensing literature.

Our analysis of the proposed method on different datasets considered here, and also its comparison with three published methods helped us in finding the answers to the questions we raised earlier in this manuscript. Our findings can be summarized as follows: 1) the promising results of the proposed method (M1) suggest that direct estimation of relative density ratios, from the data, is a viable option for supervised classification of remote-sensing time-series data; 2) the results of comparison between M1 and M2 [13] suggest that RSPRT/CUSUM statistics, when derived from the parameter time series instead of the raw data, achieve significant improvement in the performance; and 3) utilizing the benefits of parameter time series [14], [16], RSPRT statistics and relative density ratio estimation simultaneously, enabled the framework to incur lower detection delays, with higher accuracy than the rest of the methods (M2–M4) which use traditional likelihood ratios with individual densities either estimated using Gaussian kernels [13] or assumed to be Gaussian in nature [10], [14]. The difference in the results was small in case of simulated and synthetic datasets because some assumptions involved in the derivation of M2–M4 were satisfied to some extent, e.g., Gaussanity assumption, and there was no mislabeling. However, the difference became more significant in case of the real-world beetle infestation data when the Gaussanity assumption was violated and there was considerable mislabeling as well in the training data. This also suggests that the proposed framework is more robust to mislabeling as compared to the other three methods.

We also proposed a simple heuristic technique for automatic threshold tuning in near real-time change detection framework. Unlike commonly considered two indices (FP and FN), this technique considers three performance indices (FP, FN, and MD), which are challenging to deal with simultaneously while tuning the threshold. This technique proved useful in cross-validation experiments and allowed us to generate the whole set of results in a single execution, without having to select the thresholds manually in each run. The threshold values presented in Table IV were selected automatically by the framework using this technique, which verifies that it can tune the thresholds successfully.

## References

[1] G. W. Meigs, R. E. Kennedy, and W. B. Cohen, "A Landsat time series approach to characterize bark beetle and defoliator impacts on tree mortality and surface fuels in conifer forests," *Remote Sens. Environ.*, vol. 115, pp. 3707–3718, Dec. 2011.

[2] A. J. Meddens, J. A. Hicke, L. A. Vierling, and A. T. Hudak, "Evaluating methods to detect bark beetle-caused tree mortality using single-date and multi-date Landsat imagery," *Remote Sens. Environ.*, vol. 132, pp. 49–58, May 2013.

[3] A. J. Meddens, J. A. Hicke, and L. A. Vierling, "Evaluating the potential of multispectral imagery to map multiple stages of tree mortality," *Remote Sens. Environ.*, vol. 115, pp. 1632–1642, Jul. 2011.

[4] N. R. Goodwin *et al.*, "Estimation of insect infestation dynamics using a temporal sequence of Landsat data," *Remote Sens. Environ.*, vol. 112, pp. 3680–3689, Sep. 2008.

[5] N. R. Goodwin, S. Magnussen, N. C. Coops, and M. A. Wulder, "Curve fitting of time-series Landsat imagery for characterizing a mountain pine beetle infestation," *Int. J. Remote Sens.*, vol. 31, pp. 3263–3271, Jun. 2010.

[6] N. C. Coops *et al.*, "Assessing changes in forest fragmentation following infestation using time series Landsat imagery," *For. Ecol. Manage.*, vol. 259, pp. 2355–2365, May 2010.

[7] N. C. Coops, M. Johnson, M. A. Wulder, and J. C. White, "Assessment of QuickBird high spatial resolution imagery to detect red attack damage due to mountain pine beetle infestation," *Remote Sens. Environ.*, vol. 103, pp. 67–80, Jul. 2006.

[8] M. A. Wulder, J. C. White, N. C. Coops, and C. R. Butson, "Multi-temporal analysis of high spatial resolution imagery for disturbance monitoring," *Remote Sens. Environ.*, vol. 112, pp. 2729–2740, Jun. 2008.

[9] J. C. White, M. A. Wulder, and D. Grills, "Detecting and mapping mountain pine beetle red-attack damage with spot-5 10-m multispectral imagery," *BC J. Ecosyst. Manage.*, vol. 7, no. 2, pp. 105–118, 2006.

[10] J. Verbesselt, A. Zeileis, and M. Herold, "Near real-time disturbance detection using satellite image time series," *Remote Sens. Environ.*, vol. 123, pp. 98–108, Aug. 2012.

[11] J. Verbesselt, R. Hyndman, G. Newnham, and D. Culvenor, "Detecting trend and seasonal changes in satellite image time series," *Remote Sens. Environ.*, vol. 114, pp. 106–115, Jan. 2010.

[12] J. Verbesselt, R. Hyndman, A. Zeileis, and D. Culvenor, "Phenological change detection while accounting for abrupt and gradual trends in satellite image time series," *Remote Sens. Environ.*, vol. 114, pp. 2970–2980, Dec. 2010.

[13] T. L. Grobler *et al.*, "Using page's cumulative sum test on modis time series to detect land-cover changes," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 2, pp. 332–336, Mar. 2013.

[14] A. Anees and J. Aryal, "Near-real time detection of beetle infestation in pine forests using MODIS data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 9, pp. 3713–3723, Sep. 2014.

[15] A. Anees, J. Olivier, M. O'Rielly, and J. Aryal, "Detecting beetle infestations in pine forests using MODIS NDVI time-series data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS'13)*, Jul. 2013, pp. 3329–3332.

[16] A. Anees and J. Aryal, "A statistical framework for near-real time detection of beetle infestation in pine forests using MODIS data," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1717–1721, Oct. 2014.

[17] W. Kleynhans *et al.*, "Detecting land cover change using an extended Kalman filter on MODIS NDVI time-series data," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 3, pp. 507–511, May 2011.

[18] W. Kleynhans *et al.*, "Improving land cover class separation using an extended Kalman filter on MODIS NDVI time-series data," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 2, pp. 381–385, Apr. 2010.

[19] R. Watson *et al.*, *Land Use, Land-Use Change and Forestry*. Cambridge, U.K.: Cambridge Univ. Press, 2000, p. 375.

[20] L. Eklundh, T. Johansson, and S. Solberg, "Mapping insect defoliation in Scots pine with MODIS time-series data," *Remote Sens. Environ.*, vol. 113, pp. 1566–1573, Jul. 2009.

[21] Y. Fang *et al.*, "Online change detection: Monitoring land cover from remotely sensed data," in *Proc. 6th IEEE Int. Conf. Data Min. Workshops (ICDM Workshops'06)*, 2006, pp. 626–631.

[22] Y. Kang, "Real-time change detection in time series based on growing feature quantization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2012, pp. 1–6.

[23] Y. Kawahara and M. Sugiyama, "Change-point detection in time-series data by direct density-ratio estimation," in *Proc. SIAM Int. Conf. Data Min. (SDM'09)*, Sparks, NV, USA, Apr. 30/May 2, 2009, pp. 389–400.

[24] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," *Neural Netw.*, vol. 43, pp. 72–83, 2013.

[25] M. Sugiyama *et al.*, "Direct importance estimation for covariate shift adaptation," *Ann. Inst. Stat. Math.*, vol. 60, no. 4, pp. 699–746, 2008.

[26] T. Kanamori, S. Hido, M. Sugiyama, and B. Zadrozny, "A least-squares approach to direct importance estimation," *J. Mach. Learn. Res.*, vol. 10, pp. 1391–1445, Dec. 2009.

[27] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama, "Relative density-ratio estimation for robust distribution comparison," *Neural comput.*, vol. 25, no. 5, pp. 1324–1370, 2013.

[28] A. Gretton *et al.*, *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2009, ch. 8, pp. 131–160.

[29] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning for differing training and test distributions," in *Proc. 24th Int. Conf. Mach. Learn. (ICML'07)*, New York, NY, USA: ACM, 2007, pp. 81–88.

[30] T. Kanamori, T. Suzuki, and M. Sugiyama, "Statistical analysis of Kernel-based least-squares density-ratio estimation," *Mach. Learn.*, vol. 86, no. 3, pp. 335–367, 2012.

[31] T. Kanamori, T. Suzuki, and M. Sugiyama, "Computational complexity of Kernel-based density-ratio estimation: A condition number analysis," *Mach. Learn.*, vol. 90, pp. 431–460, Mar. 2013.

[32] M. Sugiyama, T. Suzuki, and T. Kanamori, "Density-ratio matching under the Bregman divergence: A unified framework of density-ratio estimation," *Ann. Inst. Stat. Math.*, vol. 64, no. 5, pp. 1009–1044, 2012.

[33] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[34] J. Im, J. Rhee, J. R. Jensen, and M. E. Hodgson, "An automated binary change detection model using a calibration approach," *Remote Sens. Environ.*, vol. 106, no. 1, pp. 89–105, 2007.

[35] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, p. 37, 1960.

[36] D. Montgomery, *Introduction to Statistical Quality Control*. 3rd ed., Hoboken, NJ, USA: Wiley, 1997.

[37] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[38] R. S. Lunetta, J. F. Knight, J. Ediriwickrema, J. G. Lyon, and L. D. Worthy, "Land-cover change detection using multi-temporal MODIS NDVI data," *Remote Sens. Environ.*, vol. 105, pp. 142–154, Nov. 2006.

[39] W. A. Shewhart and W. E. Deming, *Statistical Method From the Viewpoint of Quality Control*. New York, NY, USA: Dover, 1986.

[40] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Advances in Neural Information Processing Systems*, 2008, pp. 1433–1440.

[41] T. L. Grobler, "Sequential and non-sequential hypertemporal classification and change detection of MODIS time-series," Ph.D. dissertation, Univ. of Pretoria, South Africa, 2012.

[42] R. G. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data," *Remote Sens. Environ.*, vol. 37, no. 1, pp. 35–46, 1991.

[43] A. Stein, J. Aryal, and G. Gort, "Use of the Bradley-Terry model to quantify association in remotely sensed images," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 852–856, Apr. 2005.

[44] R. G. Pontius, Jr., and M. Millones, "Death to Kappa: Birth of quantity disagreement and allocation disagreement for accuracy assessment," *Int. J. Remote Sens.*, vol. 32, no. 15, pp. 4407–4429, 2011.

[45] A. Agresti, *An Introduction to Categorical Data Analysis*. Hoboken, NJ, USA: Wiley, 1996.

[46] A. Zeileis, F. Leisch, C. Kleiber, and K. Hornik, "Monitoring structural change in dynamic econometric models," *J. Appl. Econometrics*, vol. 20, no. 1, pp. 99–121, 2005.

[47] A. Zeileis, A. Shah, and I. Patnaik, "Testing, monitoring, and dating structural changes in exchange rate regimes," *Comput. Stat. Data Anal.*, vol. 54, pp. 1696–1706, Jun. 2010.

[48] F. Leisch, K. Hornik, and T. U. Wien, "Monitoring structural changes with the generalized fluctuation test," *Econometric Theory*, vol. 16, pp. 835–854, 2000.

[49] T. Grobler *et al.*, "An inductive approach to simulating multispectral MODIS surface reflectance time series," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 3, pp. 446–450, May 2013.

[50] B. P. Salmon *et al.*, "Unsupervised land cover change detection: Meaningful sequential time series analysis," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 2, pp. 327–335, Jun. 2011.

[51] B. Salmon *et al.*, "The use of a multilayer perceptron for detecting new human settlements from a time series of MODIS images," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 13, pp. 873–883, Dec. 2011.

[52] [Online]. Available: http://www.for.gov.bc.ca/ftp/hre/external/publish/web/bcmpb/year9/bcmpb.v9.2011kill.pdf

[53] [Online]. Available: http://www.for.gov.bc.ca/hfp/health/overview/mpb_h-istory.htm

[54] [Online]. Available: http://foresthealth.fs.usda.gov/portal/flex/ids

[55] [Online]. Available: http://landweb.nascom.nasa.gov/cgi-bin/developer/tilemap.cgi

[56] G. Jianya, S. Haigang, M. Guoruo, and Z. Qiming, "A review of multi-temporal remote sensing data change detection algorithms," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 37, no. B7, pp. 757–762, 2008.


**Asim Anees** received the B.Sc. degree in electrical engineering from NWFP University of Engineering and Technology, Peshawar, Pakistan, and the M.Sc. degree in vision and robotics (VIBOT), through European Union Erasmus Mundus Scholarship Program (Heriot-Watt University, Edinburgh, Scotland, Universitat de Girona, Girona, Spain, and Université de Bourgogne, Dijon, France). Currently, he is pursuing the Ph.D. degree in engineering at the School of Engineering and ICT, University of Tasmania, Hobart, Australia.

His research interests include image and signal processing, remote sensing, statistical detection, estimation theory, and machine learning.

Mr. Anees is a Reviewer of IEEE GEOSCIENCE AND REMOTE SENSING LETTERS (IEEE GRSL).

**Jagannath Aryal** received the Ph.D. degree in optimization and systems modeling from the Centre for Advanced Computational Solutions (C-fACS), Lincoln University, Lincoln, New Zealand, in 2010.

He is a Lecturer in Surveying and Spatial Sciences with the School of Land and Food, University of Tasmania, Hobart, TAS, Australia. His research interests include advancing the knowledge in remote-sensing discipline using geographic object-based image analysis (GEOBIA) and developing spatial models.

Dr. Aryal is in the Scientific Committee of European GIS Conference (AGILE), Australasian spatial network, and GEOBIA conference. He is a Reviewer of several remote sensing-related journals, including the IEEE and Elsevier Science Journals. He serves in the Editorial Board (Book Editor) of *Journal of Spatial Science* of Taylor and Francis Group.

**Małgorzata M. O'Reilly** received the M.Sc. degree in mathematics and education from Wroclaw University, Wroclaw, Poland, in 1987, and the Ph.D. degree in applied probability from the University of Adelaide, Adelaide, SA, Australia, in 2002.

Since 2005, she has been a Lecturer in Probability and Operations Research with the University of Tasmania, Hobart, TAS, Australia. Her research interests include stochastic modeling, Markov chains, and operations research.

**Timothy J. Gale** received the M.Eng.Sc. degree in mechatronics from the University of Melbourne, Melbourne, Vic., Australia, in 1991, and the Ph.D. degree in biomedical engineering from the University of Tasmania, Hobart, Tasmania, Australia, in 1996.

He is a Senior Lecturer with the School of Engineering and ICT, University of Tasmania, Hobart, TAS, Australia. His research interests include modeling, control, signal processing, and imaging.

Dr. Gale currently serves as a Biomedical College Board Member and Chair of the National Committee on Mechatronics for Engineers Australia.