Performance Evaluation 69 (2012) 206–226

Contents lists available at SciVerse ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

An adaptive model for online detection of relevant state changes in Internet-based systems

Sara Casolari [a,*], Stefania Tosi [a], Francesco Lo Presti [b]

[a] Department of Computer Engineering, University of Modena and Reggio Emilia, via Vignolese, 905/b, 41125 Modena, Italy

[b] Department of Computer Engineering, University of Roma, "Tor Vergata", Via del Politecnico, 1 00133 Roma, Italy

ARTICLE INFO

Article history:
Available online 13 June 2011

Keywords:
Resource management
Internet-based systems
Online statistical algorithms

ABSTRACT

Modern Internet-based systems typically involve a large number of servers and applications and require real-time management strategies for cloning and migrating virtual machines, as well as re-distributing or re-mapping the underlying hardware. At the basis of most real-time management strategies there is the need to continuously evaluate system state behavior and to detect when a relevant state change is occurring. Modern Internet-based systems open new and interesting scenarios in the field of the research on the online state change detection models.

In this paper, we propose an adaptive state change detection model that we demonstrate is suitable to analyze continuous streams of data coming from Internet-based systems characterized by high variability and non stationarity of the monitored resource measures that result in not-acceptable false alarm rates. Our model solves the limits of the traditional solutions while retaining their computational efficiency. The solution we present combines two key elements: an on-line wavelet model to denoise data streams and an adaptive detection rule. Experiments carried out using empirical and synthetic data sets confirm that the proposed method is able to signal all relevant state changes limiting the incorrect detections and to provide robust results even in non-stationary and highly variable contexts.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Most real-time management decisions related to modern Internet-based systems are activated after a notification that a relevant state change has occurred in some system resource(s). Anomaly detection, quality and access control, request redirection, process and virtual machine migration, hardware re-mapping, diagnosis and fault detection, to name a few, are examples of processes that are activated after the detection of a significant and non-transient system state change. For this reason, the most important system resources should be continuously monitored and data passed to some statistical models that decide almost immediately whether a relevant state change has occurred or not.

Internet-based systems pose novel challenges in the field of state change detection. We are moving from the more traditional area of offline contexts, where the models are applied, to well defined sets of data with (almost) no temporal constraints, to online contexts, where models receive continuous streams of data and have to answer almost immediately. We have to consider novel types of data streams. Indeed, because of internal system operations, such as context switching, virtualization, and I/O operations, and of the unpredictable variability of the user request rates, monitored resource measures are *non-stationary* and typically both the mean and the variance varyover time. Moreover, these measures exhibit what we

* Corresponding author.
E-mail addresses: sara.casolari@unimore.it (S. Casolari), stefania.tosi@unimore.it (S. Tosi), lopresti@info.uniroma2.it (F. Lo Presti).

0166-5316/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.peva.2011.05.003

can call *high variability*, that corresponds to a standard deviation comparable to (and even larger than) the mean and that can be quantified by a coefficient of variation larger than one. Finally, non-stationary behaviors occur at a faster time scale with significant variations even over short time intervals.

In these contexts, the most common models used to support the detection of relevant state changes, such as Particle Filtering [1], Kalman filter [2] and Sequential Monte Carlo Method [3], are not effective because they require a knowledge of some statistical characteristics of the time series [4,5]. Other popular state change detectors, such as the threshold-based detectors, the Shewhart chart, the Exponential Weighted Moving Average (EWMA) chart and the traditional Cumulative Sum (Cusum) chart, are often inadequate because when data streams are characterized by non-stationary behaviors and high variability, they cause either a large number of false detections or an absence of detections depending on the chosen parameters of the model [6,7].

In this paper, we propose a new adaptive model that combines two key ingredients:

- a *detection rule* based on a new adaptive implementation of the Cusum model [8] that is able to keep the detector performance (in terms of both false detection rate and absence of detection rate) close to optimal;
- a *data representation* that uses an online wavelet-based filter that is able to rectify the monitored data by eliminating random non-Gaussian errors while retaining the main features of the original data stream.

We demonstrate that the proposed adaptive state change detection model is able to provide the best results for several statistical characteristics of data coming from real and emulated contexts, and that it is able to satisfy the temporal constraints that are typical of real-time resource management systems.

The rest of the paper is organized as follows. In Section 2 we present the related work and compare our contribution with the state of the art. Section 3 presents the definition of relevant state change detection applied to continuous streams of data, and gives additional motivation for this research. Section 4 details our adaptive detection model, which is compared to other detection algorithms in Section 5. Section 6 introduces the main performance metrics used to evaluate the detection quality and compares the results of the proposed adaptive model against other popular models for several data streams. Section 7 analyzes the results of the proposed model for real data sets. We conclude the paper in Section 8 with some final remarks.

2. Related work

This paper differs from most literature on state change detection that is oriented to offline models. Instead, we are interested to online detection algorithms that analyze continuous data streams as required by typical modern Internet-based systems, where multi-core architectures host multiple interactive services. Detecting changes through offline models on the entire data set has several advantages, among which the possibility of evaluating its statistical characteristics and the possibility of adopting complex stochastic models and algorithms because there are no temporal constraints imposed by the application context. Online models represent a time series as a stream of samples, whose statistical properties (mean, variance, etc.) vary over time. Since it is not possible to predict how the statistical properties of the data stream will evolve, all the models that require a statistical characterization of the data are not applicable in this context. On the other hand, more advanced models that are able to adapt to changes in the statistical properties of the data are not suitable because their high computational complexity is not compatible with the time constraints of online state change detection. Moreover, most online models do not work well when the data are characterized by non-stationarity and high variability. In general, these properties limit the performance of the online models that work directly on the raw data coming from the data streams [9]. For this reason, the majority of papers on the online models working on non-stationary and highly variable data streams [10–12] combine two elements: a model for the data representation that has to provide a denoised/rectified view of the raw data, and a model for the detection rule that has the scope to detect when a change happens. For this reason, we present separately the data representation and the detection rule problems that are at the basis of the algorithms for online state change detection [7,13].

The majority of data representation techniques assumes a preliminary knowledge about the statistical properties that characterize the state of the time series and their time evolution (e.g., Kalman filter [2], sequential Monte Carlo method [3]) or an empirical evaluation of them (e.g., particle filtering [1]). The non-stationary and unpredictable behavior that characterizes the data streams coming from the monitored resources of Internet-based systems prevents the applicability of these models because of the impossibility of defining *a priori* a set of statistical properties that are able to represent the evolution of the entire time series. Moreover, a continuous estimation of the statistical properties is incompatible with online constraints. The simple models proposed in [12,14] based on a Kalman filter for data representation are able to support the non-stationarity of the Internet-based systems because they do not require any *a priori* information about the time series states, but they are unable to provide reliable data representations in highly variable with variable variance contexts. Other popular methods to remove perturbations are based on linear filtering methods, such as mean filtering, exponential smoothing [6,9,11] and Fourier transform [15]. They are adequate for online applications because of their low computational cost. In highly variable contexts, the simplicity of these methods has some drawbacks: the linear filtering methods remove all frequencies above a cutoff value, hence if the resulting smoothed representation removes all the perturbations, significant features of the time series, such as relevant stage changes, also might be smoothed out. On the other hand, if the resulting data representation preserves the main original features, many perturbations may continue to characterize the time series and risk to cause many false detections.

Therefore, we think it is necessary to refer to non-linear and multiscale data representation algorithms that maintain a computational complexity compatible to online contexts. Our choice goes to the wavelet-based algorithm [16] because it is able to separate the main features of the time series from its perturbations. The wavelet-based methods are applied to many fields [17,16] because they are able to limit the time series perturbations also in highly variable contexts. The reason is that they use an orthonormal basis localized in space and frequency. On the other hand, the linear filters, such as exponential smoothing, use the sine-cosine basis that is localized only in frequency and not in space. In this paper, we propose an online adaptation of the wavelet-based representation described in [18] that considers a moving window of dyadic length and a dynamic estimation of the model parameters.

We pass now to consider the detection rule problems. The online constraints prevent us from adopting several algorithms and models for state change detection working offline (e.g., [8]). Moreover, we cannot refer to proposals (e.g., machine learning, Principal Component Analysis, neural networks [19]) that assume some prior knowledge about the time series characteristics, especially about the probability of state changes and their distribution, because the considered scenarios are characterized by non-deterministic, non-stationary and highly variable behaviors.

When it is impossible to assume or evaluate online the statistical properties characterizing a time series, a typical approach for detecting state changes is to refer to threshold-based algorithms [20]. The results of these models are not robust because there is no theoretical support for the choice of the threshold value(s), that remains an empirical choice. Although these models are unsuitable for the highly variable context of interest for this paper, we consider a threshold-based algorithm just for comparison purposes.

In this paper, we refer to the family of Cusum models that are considered the best choice for the online state change detection [6]. To provide optimal results, the Cusum model requires us to know the reference value of the system state. However, when the state value is unknown, such as in our contexts, the Cusum is applied to a data representation based on exponential smoothing [6,7] that provides a dynamical estimation of the state. In [21] we use this technique on a modified version of the Cusum that is able to adapt its parameters on the basis of the process variance. For this reason, we consider a Cusum-based statistical model as a basis for our detection rule. However, we found out that a detection algorithm alone it is not sufficient because, in contexts characterized by highly variable time series with variable intensity of perturbations, the data representation based on the exponential smoothing exhibits several problems and the performance of the entire state change detection algorithm tends to decrease.

The main strength of our approach, that differentiates the proposed online state change detection model from all the previous works, is the ability to reliable state change detections in non-stationary and highly variable contexts. This critical task is carried out by a novel adaptive detection model that uses a data representation based on an adaptive version of the online wavelet transform and an adaptive version of the Cusum detection rule. Our proposed adaptive detection model is the first to achieve a good tradeoff between the number of false detections and the absence of detections in highly variable contexts because it uses a data representation that is able to remove all undesired perturbations while preserving the most relevant data information, such as the state changes.

3. Problem definition

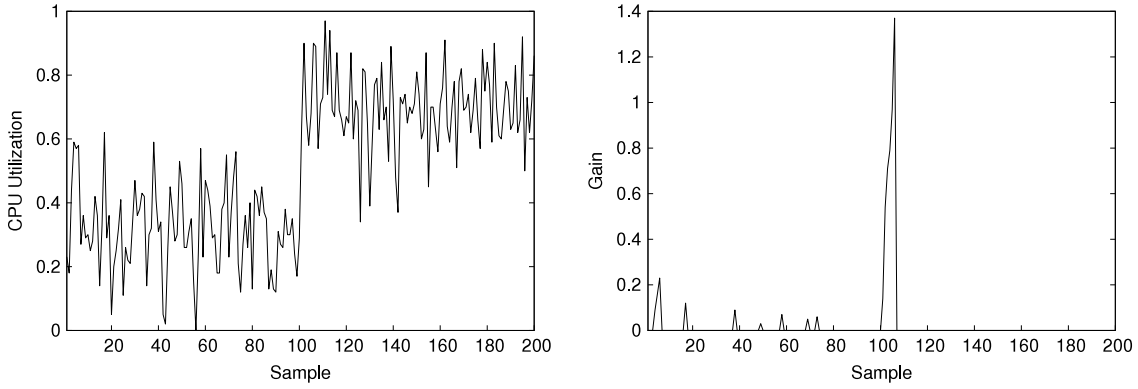
3.1. Online detection problem

We are interested in an online version of the state change detection problem where the time series is represented by a continuous stream of measurements of the resource usage of Internet-based servers.

In time series characterized by non-stationary conditions, a *state* is defined as a subset of the continuous data stream where data are characterized by the same statistical properties that we generically denote by ϑ. It can refer to one attribute (e.g., mean, variance, distribution) or a combination of them. We define *relevant state change* as a shift larger than $\Delta > 0$ in the ϑ metric that the model should be able to timely detect. The actual value of Δ is actually application-dependent.

We consider a continuous stream of temporally ordered samples $\{y\} = \{y_1, \ldots, y_i, \ldots\}$, and we focus on operations and parameters that the model has to apply at a generic sample y_i. Before this sample, we can assume that the model has identified one or more state(s), each characterized by an online computation of the statistical characteristics $\widehat{\vartheta}$ of interest for the detection model. Let us denote by y_s and by $\widehat{\vartheta}_s$ the first sample of the present state of the data subset including y_i and its estimated statistical characteristics, respectively. (We implicitly assume that there was no relevant state change between y_s and y_i.) At the sample y_i, the online detection model has to evaluate whether a relevant state change occurs or not. To do this, a statistical representation $f_y(y_s, \ldots, y_i)$ of the samples between y_s and y_i is dynamically evaluated to provide an estimation of the statistical properties of the data stream and a *detection rule* must verify whether the deviation among the estimated statistical properties of the current state $\widehat{\vartheta}_s$ and the statistical representation $f_y(y_s, \ldots, y_i)$ of the samples between y_s and y_i overcomes a certain threshold. This is typically done on the basis of the likelihood ratio [22], that estimates the deviations among the statistical representation of the data samples and the current state.

We distinguish two classes of detection models, where the differences reside mainly in the way to consider these deviations, the way to choose the statistical threshold, and the way to represent the samples between y_s and y_i. In particular, the first class uses as a threshold a function f^1_Δ of the minimum value Δ of the shift that the model should capture and compares $f^1_\Delta(\Delta)$ with a punctual evaluation of the deviation between the estimation of the current state $\widehat{\vartheta}_s$ and the function



(a) Time series. (b) Gain function$_g$.

Fig. 1. Detecting relevant state changes through a gain function.

$f_y(y_s, \ldots, y_i)$. The models belonging to this class (e.g., the threshold-based models [20] and the Exponential Weighted Moving Average (EWMA) [6]) can detect the presence of a relevant state change on the basis of the following equation:

$$\text{detection rule} = \begin{cases} \text{state change}, & \text{if } |\widehat{\vartheta}_s - f_y(y_s, \ldots, y_i)| \geq f_{\Delta}^1(\Delta) \\ \text{no state change}, & \text{otherwise.} \end{cases} \tag{1}$$

The second class of state change detection algorithms are based on a *gain function* g (e.g., the Cumulative Sum [8]). The gain function g can be considered as an accumulator of the detection model: during a relatively stable state it should be close to zero, and it should depart from zero when a relevant change occurs in the time series. It sums up the consecutive partial increments (decrements) among the online estimation of the current state $\widehat{\vartheta}_s$ and the sample y_i every time that the deviation between the current state estimation $\widehat{\vartheta}_s$ and the online data representation $f_y(y_s, \ldots, y_i)$ reaches a reference value of the minimum shift Δ computed by the function f_{Δ}^2. In such a case:

$$g_i = g_{i-1} + f_g(|\widehat{\vartheta}_s - y_i|), \quad \text{if} \quad |\widehat{\vartheta}_s - f_y(y_s, \ldots, y_i)| \geq f_{\Delta}^2(\Delta) \tag{2}$$

$$g_i = g_{i-1}, \quad \text{if } |\widehat{\vartheta}_s - f_y(y_s, \ldots, y_i)| < f_{\Delta}^2(\Delta) \tag{3}$$

where g is initialized to $g_0 = 0$ and it is reset to 0 every time that a state change is detected. g is characterized by a stable value when the condition $|\widehat{\vartheta}_s - f_y(y_s, \ldots, y_i)| \geq f_{\Delta}^2(\Delta)$ is not satisfied. The function $f_g(|\widehat{\vartheta}_s - y_i|)$ computes the actual value of increment (decrement) among the online estimation of the current state $\widehat{\vartheta}_s$ and the actual sample y_i that must be summed to g_{i-1}. The choice of the function f_g differs from a detection model to another [22] and depends on the statistical characteristics representative of a time series. When the intensity of consecutive detected shifts is over a given threshold H, the model has to signal a relevant state change. Therefore, the detection rule of this class of models can be written as follows:

$$\text{detection rule} = \begin{cases} \text{state change}, & \text{if } g_i \geq H \\ \text{no state change}, & \text{otherwise.} \end{cases} \tag{4}$$

The choice of the characteristic threshold H depends on the specific gain function g and on the performance required by the state change detection algorithm.

We give an example of gain function computed on the time series of Fig. 1(a). It refers to the CPU utilization of a server sampled every five seconds and that is characterized by a relevant load change at sample 100. Fig. 1(b) reports the results of the gain function g applied to that time series. As expected, g is characterized by a significant increment at the instant of the relevant change in the system load.

For both classes of detectors, when a relevant state change is detected, the detection model has to provide an estimation of the statistical characteristic $\widehat{\vartheta}$ of the novel state. For simplicity reasons, without loss of generality, in this paper we consider the state change detection problem applied to the case of relevant changes in the mean of the time series. Hence, we consider $\widehat{\vartheta} = \mu$. In this way, after a state change the estimation of novel state should be able to capture the central tendency of the time series during that state.

3.2. Online detection problem in non-stationary, highly variable and spike contexts

The performance of the state change detection algorithms depends heavily on the statistical properties of the time series [6]. The context of the Internet-based servers where data streams are continually collected from monitored system resources opens interesting new challenges. These servers are subject to variable resource demands, heterogeneous processes, and different levels of virtualizations. Consequently, monitored system resources are characterized by high levels

of utilization, unexpected and unpredictable events and several sources of perturbations which result in spikes. The highly variable, non-stationary and unpredictable usage of resource measures and the presence of spikes and other perturbations with variable intensities complicate the identification of the system state and state changes. Detecting relevant state changes is further complicated by the temporal constraints imposed by real-time management strategies that receive the outputs of the online detection algorithms. This novel perspective prevents the application of state of the art algorithms working offline (e.g., [8]). We will show that it causes poor performance even of the most popular algorithms that can be applied to online continuous streams.

We give some examples of the major issues that affect the online detection algorithms when applied to Internet based servers. The architecture that we use for our evaluations is a typical distributed multi-tier Web system that is based on the implementation presented in [23]. The application servers are deployed through the Tomcat servlet container, and are connected to MySQL database servers. In our experiments, we exercise the system through realistic traces. We emulate more variable scenarios by means of the TPC-W workload generator [24]. Our workload scenarios are described by distinct step variations of the external load obtained by changing the number of emulated browsers. The algorithm that we use in the following evaluation is the Cumulative Sum (Cusum) that is a widely used sequential analysis technique for detecting changing points in time series.

In the following examples, we use the CPU utilization of the database server as the representative system resource measure, because of the strong correlation of its utilization level with the external load (larger than 0.8). We set as representative state values $\{\mu\}$ the mean of the CPU utilization samples during the period in which the external load was constant and select the samples of change $\{s\}$ as the sample in which the number of emulated browser changes. Fig. 2(a), Fig. 3(a) and Fig. 4(a) show the CPU utilization of three database servers that are subject to different external loads: the first user scenario is characterized by relevant changes in the number emulated browser at samples {50, 150, 200, 250, 315, 440, 475, 540}; the second one by any relevant change; and the last one by a single relevant change, at sample 300, in which both the user requests type and the number of emulated browser are changed. We consider that all detected changes signaled correctly by the detection algorithm are called *true positives* (*TP*). If an algorithm does not detect one relevant state change, the related sample is classified as *false negative* (*FN*). Analogously, the detection of a change is classified as *false positive* (*FP*) if it occurs when the time series is in a stable state.

Fig. 2(b), Fig. 3(b) and Fig. 4(b) show the results of the gain function g function referring to the Cusum algorithm. In all of these figures, the horizontal dotted lines represent the upper and lower H thresholds set by the model, that signals a relevant state change every time the gain function g overcomes these limits. Each circle at the bottom and at the top of the figures denotes a false positive detection, that is, a signaled change that does not correspond to a real state change. A cross denotes a correct detection of a state change, that is a true positive.

Fig. 2(a) reports a time series characterized by non-stationary, highly variable and unpredictable data. These statistical properties of the data stream make it difficult for the model to identify intervals of stability in which the statistical properties ϑ remain constant. This reflects in oscillating values of the gain function that cause a lot of signals not corresponding to real state changes: there are eleven false positive detections and two false negative detections that are the undetected changes at samples 315 and 540.

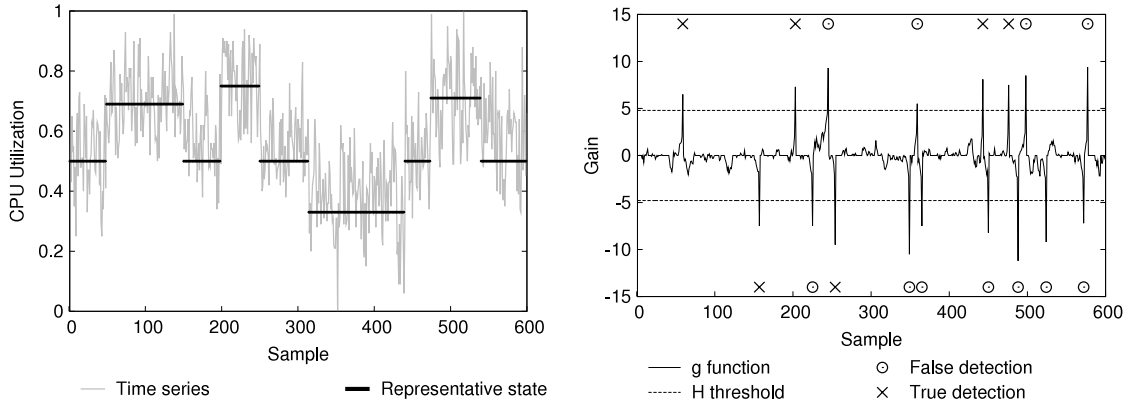
Fig. 3 reports a typical result achieved by existing algorithms by the Cusum algorithm applied to spiky time series. Since the g function accumulates the deviations of the monitored data from the current state, it records also the contributions of instantaneous spikes and anomalous perturbations departing from the current data behavior. These contributions are usually of high intensity and cause an immediate exceeding of the H threshold. For this reason, we have false positive detections corresponding to the peaks of the time series. Removing out-of-scale values through filtering algorithms before applying the detection rule should solve this problem. For this reason, instead of using the original time series $\{y\}$, we suggest the use of an online *data representation* $\{x\}$, that is a rectified version of the monitored samples. More details will be given in Section 4.3.

In Fig. 4 we report an example where the Cusum detection model is applied to a time series characterized by a variable variance that passes from a higher value in the interval [0:300] to a lower value after the state change. Here, the detector exhibits two problems: false positive detections at samples 90 and 112 during the interval of high variance and excessive delay because the relevant state change is signaled 21 samples after its occurrence at sample 300. A late detection is equivalent to an incorrect detection when the delay is incompatible with the temporal constraints imposed by the online context. These problems are the consequence of a static estimation of the time series variance used by the models for detecting changes in the mean [7]. For this reason, an efficient state change detection model in the context of Internet-based systems must adapt its detection policies to the variable variance of the time series.

This preliminary analysis shows the main problems that may affect an online state change detection algorithm when the time series shows the main statistical properties characterizing data streams related to Internet-based servers: false detections and lack of signals. We conclude that novel online detection models are required in the context of data streams related to modern Internet-based architectures.

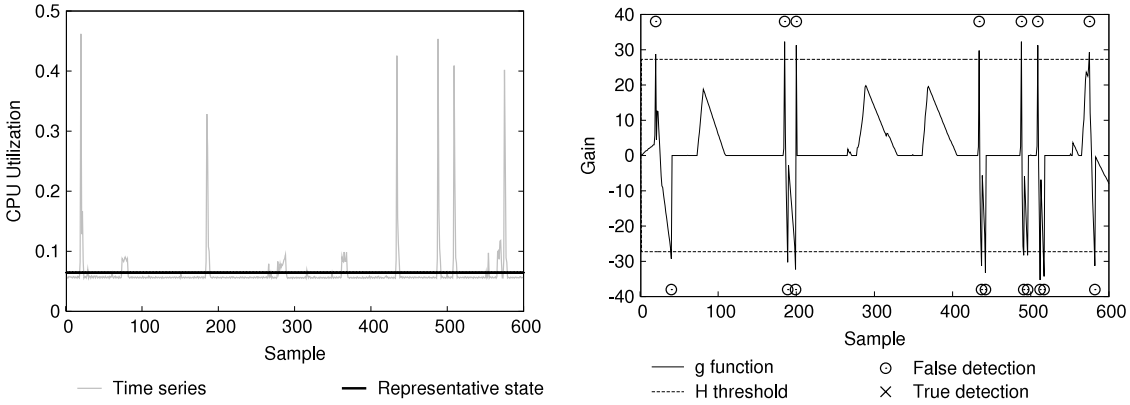
4. Adaptive model for state change detection

The performance of change detection algorithms is highly dependent on the statistical characteristics of the measured data [6]. Because of the inherent variability and non-stationary behavior of the monitored processes related to Internet-



(a) Time series. (b) Results of the Cusum algorithm.

Fig. 2. Time series characterized by non-stationarity, high variability and unpredictability.



(a) Time series. (b) Results of the Cusum algorithm.

Fig. 3. Time series characterized by spikes.

based servers, existing algorithms achieve poor results. We propose a new methodology that extends and integrates two well known models (Cusum [8] and wavelet [16]) in two directions: adaptability and application to continuous data streams. The integrated version can be efficiently implemented and is therefore suitable to online detections. The motivation for our choice comes from the observation that the Cusum change detection algorithm has been proven to be optimal in terms of detection delay and minimum false alarm probability [25]. Nevertheless, its direct application leads to poor results when the data stream is characterized by high variability. To this end, we combine it with an online wavelet filter. We adopt wavelet filters for their ability to remove random errors from time series (an operation which is known as denoising or rectification depending on the context [18]) without affecting the relevant features of the original data stream. In this respect, they have proven to be optimal with respect to various error norms and smoothness property [18].

4.1. The baseline Cusum algorithm

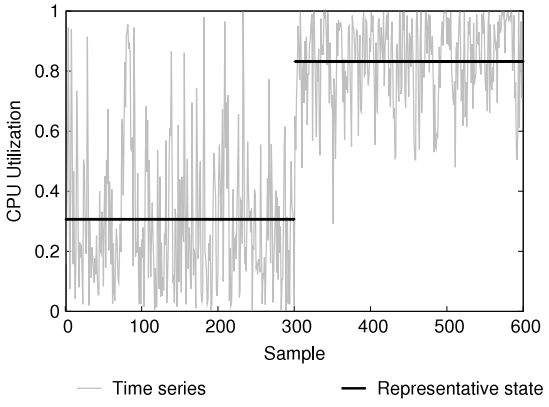
Cusum is a widely used model for detecting changing points in otherwise stationary time series with known statistical characteristics. Proposed by Page in 1954 [8], this technique has been extensively studied and extended since [4,26]. The Cusum algorithm has been shown to be optimal when the variance of the time series does not change, in that it guarantees a minimum mean delay to detection in the asymptotic regime when the mean time between false alarms goes to infinity [25].

In this section, we describe the baseline Cusum algorithm; we use it in the experiments as a comparison testbed and to illustrate the benefits of our online Adaptive Cusum algorithm.

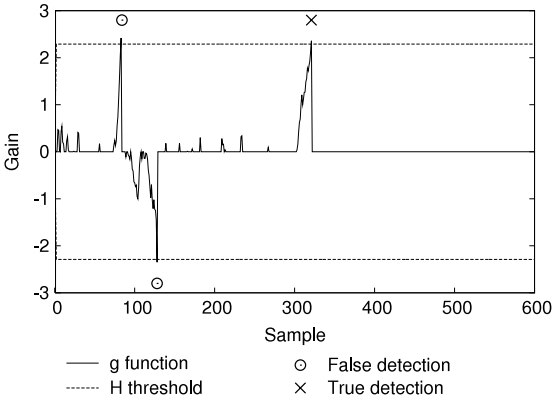
Given a time series $\{y_s, \ldots, y_i\}$ with known mean μ_s, the one-sided Cusum detects an increase in the mean through the following g_i gain function:

$$g_0^+ = 0 \tag{5}$$

$$g_i^+ = \max\{0, g_{i-1}^+ + y_i - (\mu_s + K^+)\} \tag{6}$$



(a) Time series.



(b) Results of the Cusum algorithm.

Fig. 4. Time series characterized by variable variance.

which measures positive deviations of the monitored time series y_i from the state value μ_s. The gain function g_i^+ accumulates deviations of y_i from the state value μ_s that are greater than a pre-defined threshold K^+, and resets to 0 on becoming negative. The term K^+, which is known as the allowance or slack value, determines the minimum deviation that the statistics g_i^+ accounts for, and depends on the choice of the minimum shift to be detected, Δ. According to the detection rule common to all detection models, a positive change is signaled when g_i^+ exceeds a design chosen threshold H^+.

The one-sided Cusum test for detecting negative deviations is defined similarly, as:

$$g_0^- = 0 \tag{7}$$

$$g_i^- = \max\{0, g_{i-1}^- + (\mu_s - K^-) - y_i\} \tag{8}$$

A negative change is signaled when g_i^- exceeds a threshold H^-.

A two-sided test to detect both increases and decreases is obtained by applying the two tests simultaneously. As in this paper we are interested in detecting both increases and decreases, we will consider the two-sided test. For the sake of simplicity we will consider the symmetric case whereby $K^+ = K^- = K$ and $H^+ = H^- = H$.

When a relevant shift is detected, the Cusum test also provides an estimate of the new system state μ_{s+1} through the following equations:

$$\mu_{s+1} = \begin{cases} \mu_s + K + \dfrac{g_i^+}{N^+} & \text{if } g_i^+ > H \\ \mu_s - K - \dfrac{g_i^-}{N^-} & \text{if } g_i^- > H \end{cases} \tag{9}$$

where $N^+(N^-)$ denotes the number of samples elapsed since the last time $g_i^+(g_i^-)$ was set to zero, that is $N^+ = i - \inf\{j \mid g_j^+ = 0\}$ and similarly for N^-.

The performance of the Cusum test is expressed in terms of the so-called *Average Run Lengths* (ARL): ARL_0 denotes the average number of samples between false detections when no change has occurred; ARL_1 denotes the average number of samples to detect a change when it does occur. Ideally, ARL_0 should be large, while ARL_1 should be small. Both ARL measures are affected by the design parameters H and K. To achieve good performance, the suggested values in stationary time series are $K = \frac{\Delta}{2}$, where Δ is the minimum shift to be detected, and $H = 5\sigma_y$, where σ_y is the time series standard deviation [6]. In these conditions characterized by stationarity and constant variance of time series, the choice of $K = \frac{\Delta}{2}$ has been shown to provide near minimal ARL_1 for a wide range of threshold values H, while $H = 5\sigma_y$ guarantees $ARL_0 = 470$ for a shift of $\Delta = \sigma_y$, which is typically considered as a reference value.

There are several techniques to compute ARL_0 and ARL_1. In this paper, we consider the Siegmund approximation that combines simplicity and efficacy [27].

In Fig. 5 we plot ARL_0 and ARL_1 ($\Delta = \sigma_y$, $K = \Delta/2$). As shown in this figure, the performance of the Cusum algorithm is heavily dependent on the ratio H/σ_y. Let us consider first a fixed standard deviation σ_y and let us vary the threshold value H. As expected, for larger values of the threshold H, it is more difficult to incur in false detections (that is, it is more difficult that perturbations of g_i exceed the threshold) at the cost of an increasing detection delay ARL_1 since g_i needs to attain larger values for detection. From the same figure, we can also observe that, by increasing the threshold H, the false detection rate (which is inversely proportional to ARL_0) decreases exponentially fast at the expense of higher ARL_1.

Let us now fix H and let us vary the standard deviation σ_y. The key observation here is that the performance of the Cusum rapidly degrades as σ_y increases (smaller values of H/σ_y), since the false detection rate increases exponentially fast. Observe that this could be compensated by using a large value of H which, on the other hand, has a negative impact on detection

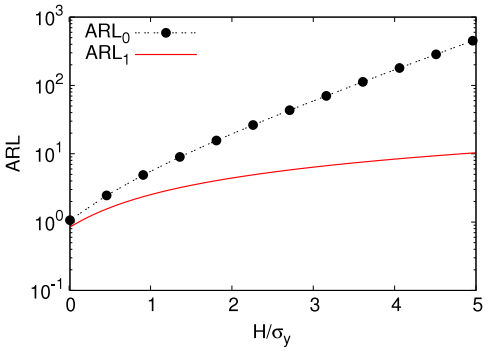


Fig. 5. Average run length, $\Delta = \sigma_y$.

delay which, in on-line operation, must be kept as small as possible. As a consequence, in a non-stationary context whereby the variance can exhibit significant variation over time, we should not rely on a fixed set of parameters.

4.2. The online adaptive Cusum detector

The basic Cusum algorithm is the best choice for statistical quality control of many processes characterized by stationary time series [6]. Unfortunately, it cannot be directly applied to online state change detection in Internet-based systems. First of all, the Cusum algorithm requires knowledge of the stationary state value μ_s against which measuring the time series deviations. Our experience shows this does not apply to computer system resource dynamics: CPU, disk and network usage dynamics are clearly non-stationary with respect to all relevant statistics, where the mean value and standard deviation vary over time. Second of all, the non-stationary behavior prevents the possibility of setting the design parameters H and K that can guarantee good performance over time. This is a critical issue because, as we have seen, for any fixed value of H the Cusum performance rapidly degrades as the time series standard deviation σ_y increases.

In this section, we propose a version of the Cusum model that we call *Adaptive Cusum*, capable of detecting state changes in face of the non-stationary characteristics of the continuous data stream. The proposed algorithm aims to solve the limitations of the baseline Cusum algorithm outlined above as follows:

(1) we replace the reference state value μ_s by an adaptive estimation μ_i of the time series mean;
(2) we replace the standard deviation σ_y by an online estimation σ_i of it ;
(3) we dynamically adjust the threshold H as to provide a target ARL_0 performance in spite of high and possibly time varying variances.

The proposed Adaptive Cusum-based detector consists of two components: an Exponential Weighted Moving Average (EWMA) filter that tracks the slow varying mean, and a two-sided Cusum test with varying thresholds for detecting relevant state changes. We consider the following tracking EMWA filter:

$$\mu_i = \alpha y_i + (1-\alpha)\mu_{i-1} \tag{10}$$

where $0 < \alpha \leq 1$ is typically set to $1/(1 + 2\Pi * cf)$ and cf is the cutoff frequency of the EWMA filter [28].

The time series variance σ_y is in general unknown and varying over time. We resort to a widely adopted approximation of variance that, for the sake of simplicity necessary in an online context, basically replaces the standard deviation σ_y with the mean deviation $E[\mid y - E[y] \mid]$ computed over time [29]:

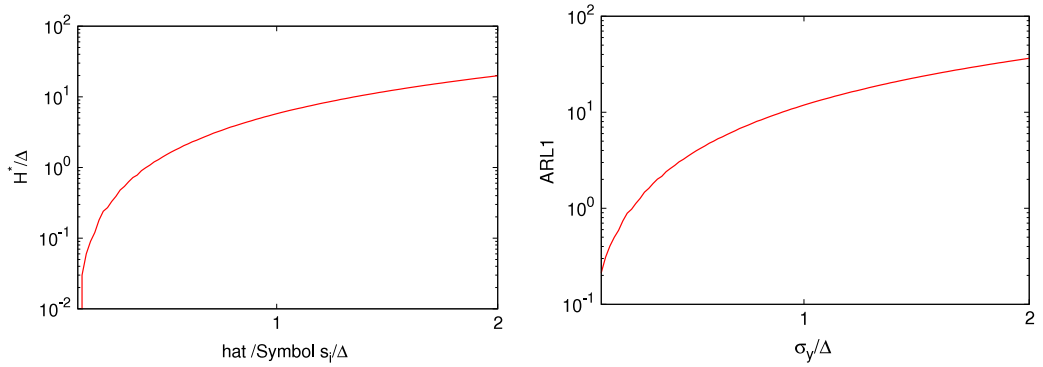
$$\sigma_i = \alpha \mid y_i - \mu_i \mid + (1-\alpha)\sigma_{i-1} \tag{11}$$

where $0 < \alpha \leq 1$ is the same as in (10).

By setting a suitable value for ARL_0 so to guarantee good performance in terms of low false detections, we are able to dynamically adjust the value of the threshold H^* to reflect the variation of the data stream variance. This keeps a desired performance of the state change detector. The computation of H^* for the desired ARL_0 is carried out by a numerical inversion of the Siegmund approximation [27], where the parameters are set as following: $K = \frac{\Delta}{2}$, where Δ is the smallest shift we want to detect (state changes smaller than Δ are accounted for by μ_i which tracks the state). Hence, to evaluate H^* we use:

$$ARL_0^{+/-} = \frac{e^{\frac{\Delta}{\sigma_i}\left(\frac{H^*}{\sigma_i}+1.166\right)} - \frac{\Delta}{\sigma_i}\left(\frac{H^*}{\sigma_i}+1.166\right) - 1}{\frac{\Delta^2}{2\sigma_i^2}}. \tag{12}$$

The resulting curve is plotted in Fig. 6(a) where the target run length is set to $ARL_0 = 1000$. As expected, H^* increases significantly as a function of σ_i. We can compute a closed form approximation of the curve through a polynomial



(a) Adaptive threshold H^*. (b) ARL_1.

Fig. 6. Adaptive threshold H^* and average Run Length ARL_1 as a function of the time series variance (target: $ARL_0 = 1000$).

approximation. For $ARL_0 = 1000$, we obtained a good approximation with the following 3rd degree polynomial:

$$H^*(\sigma_i) = -1.3645\sigma_i^3 + 10.3031\sigma_i^2 + 3.1860\sigma_i - 0.2882 \tag{13}$$

The detection rule is provided by modifying (6) and (8) as following:

$$g_0^+ = 0 \tag{14}$$

$$g_i^+ = \max\{0, g_{i-1}^+ + y_i - (\mu_i + K^+)\} \tag{15}$$

$$g_0^- = 0 \tag{16}$$

$$g_i^- = \max\{0, g_{i-1}^- + (\mu_i - K^-) - y_i\} \tag{17}$$

where μ_s is replaced by the adaptive estimation of the state μ_i. A state change is detected whenever g_i^+ or g_i^- exceeds the threshold H^*. When a change occurs, the online estimation of the state μ_i is updated through the online implementation of (10):

$$\mu_i = \begin{cases} \mu_{i-1} + K + \dfrac{g_i^+}{N^+} & \text{if } g_i^+ > H^* \\ \mu_{i-1} - K - \dfrac{g_i^-}{N^-} & \text{if } g_i^- > H^* \end{cases} \tag{18}$$

This adaptive version of the Cusum algorithm we have presented is able to deal with non-stationary time series. It does so by adapting the threshold H^* to the online estimate of the time series standard deviation, to guarantee acceptable level of false detections. Clearly this comes at a cost. In Fig. 6(b) we also plot the ARL_1 of the Adaptive Cusum as a function of σ_i. ARL_1 is a function of H^*, which in its turn, in our scheme, is a function of σ_i. As expected, ARL_1 grows for increasing values of σ_i. This can be explained by observing that for higher values of σ_i the algorithm requires higher values of the threshold H^* to guarantee a desired level of false detection rate. The higher values of the threshold H^* translate into higher detection delays (see Fig. 5).

As confirmed by our experiments, because of the high variance that characterizes time series related to Internet-based servers, the Adaptive Cusum algorithm alone might not be sufficient to guarantee good detection performance. To overcome these limitations, we combine our novel change detection algorithm with a data representation obtained by the *rectification* wavelet algorithm that should reduce the noise/perturbations from the measurements and improve the change detection performance. We discuss it below.

4.3. Wavelet-based denoising

The performance of the adaptive Cusum algorithm is negatively affected by high variability in the original time series. Hence, we have to reduce the amount of noise/perturbation through some rectification algorithm. There are many existing techniques which can be broadly classified into linear and non-linear methods. Linear filter-based methods are widely adopted for their ease of use and computational efficiency. Unfortunately, they are single scale by nature, that is, they are simple low pass filters. As a consequence, if a time series contains features at multiple scales, linear filters must tradeoff the extent of noise removal with the quality of the retained features [17]. In our context, this would either result in reduced noise removal and too many false detections or in excessive smoothing which would nullify the effectiveness of the overall change detection scheme. Non-linear methods, inherently multiscale, exhibit better performance than the linear counterparts. In

particular, wavelet-based methods have been shown to be nearly optimal for various error norms and smoothness of the resulting time series [18]. Wavelet denoising exploits the use of a orthonormal basis localized both in space and frequency which allows us to reduce/remove noise without smoothing the time series features.

In this section we describe the rectification phase we adopt before the application of our adaptive state change detection model. In other words, to improve the performance of the change detection algorithm, we find it better to consider a *rectified data representation* $\{x\}$ [17] than the original data stream $\{y\}$. Here, $\{x\}$ retains the significant features of the original data but removes (most of) the variability that can be ascribed to short-term perturbations. We regard $\{x\}$ as the data representation of the monitored process obtained through an online version of the wavelet model, and we apply state change detection rules based on the adaptive version of the Cusum algorithm to the rectified time series $\{x\}$. We refer to an online version of the wavelet-based denoising/rectification [17], that is a powerful method for filtering/rectification, and use it as a basis of our online data representation. Rectification based on the wavelet transform [16] is able to isolate and remove the perturbations that affect the monitored processes. As observed above, the reason for this result is that it uses an orthonormal basis localized in space and frequency while the traditional solutions based on exponential smoothing techniques work only in the frequency domain.

We start giving a definition of the wavelet transform, commonly used in offline contexts, that represents a time series as the sum of a shifted and scaled version of a base wavelet function ψ and a shifted version of a low-pass scale function ϕ. With a proper choice of the wavelet and scale functions, the resulting families of functions are:

$$\psi_{mk}(n) = \sqrt{2^{-m}}\psi(2^{-m}n - k) \tag{19}$$

$$\phi_{mk}(n) = \sqrt{2^{-m}}\phi(2^{-m}n - k) \tag{20}$$

where m and k are the dilation and translation parameters, respectively, from an orthonormal basis. A time series $\{y\}$ can be conveniently rewritten as follows:

$$y_i = \sum_{k=1}^{n2^{-L}} a_{Lk}\phi(i) + \sum_{m=1}^{L}\sum_{k=1}^{n2^{-m}} d_{mk}\psi_{mk}(i) \tag{21}$$

where a_{Lk} is the k-th scaling function coefficient at the coarsest scale L, d_{mk} is the k-th wavelet coefficient at scale k, and n is the time series length. The coefficients m and k are computed by the inner product of $\{y\}$ with the base functions. Computation of the transform and its inverse can be done in $O(n)$. As indicated in [17], in our implementation we set the coarsest scale L equal to 5 if the time series is perturbed by white noise, $L = 4$ otherwise. A key feature of this representation is that the wavelet decomposition captures significant signal features in a few relatively large coefficients, while perturbations result uncorrelated. As a result, perturbations – and perturbations only – can be effectively removed by setting equal to zero the wavelet coefficients smaller than a threshold specified below.

Summing up, we obtain the data representation $\{x\}$ of the original time series $\{y\}$ through the following steps:

(1) compute the wavelet transform of the original time series $\{y\}$. We use the standard Haar function [30] as a base wavelet, which consists of a simple rectangular impulse function;
(2) set to zero the wavelet coefficients which are lower than a suitable threshold t_m where m is the dilation parameter. As indicated in [17], we set the threshold $t_m = \sigma_m\sqrt{2\log n}$ where $\sigma_m = \frac{1}{0.6745}\text{median}\{|d_{mk}|\}$;
(3) compute the inverse wavelet transform to obtain $\{x\}$.

This rectification technique has been proved to be superior to other approaches [18] but it can only be applied on offline operations. Since this would be not acceptable for real-time operations, we consider the online version proposed in [17] where, at each step i, the rectified value x_i is computed using only past values as follows:

(a) consider the sub-sequence $\{y_i\} = (y_{i-M+1}, \ldots, y_i)$ of maximum dyadic length, e.g., with $M = \lfloor \log_2 i \rfloor$;
(b) compute the rectified sequence $(z_1, \ldots, z_M)$ of the sub-sequence $\{y_i\}$ using steps 1–3 above;
(c) set $x_i = z_M$, e.g., set the actual rectified value equal to the last value of the rectified sequence computed at step (b).

This online version can be computed in $O(n \log n)$ steps (see [17] for details) and it is therefore suitable to efficient implementations that are required by real-time management systems.

The overall scheme which combines the wavelet-based rectification phase described in this subsection and the adaptive change detection phase described in Section 4.2 is summarized in Fig. 7. The original measurements $\{y_i\}$ are fed to the rectification phase which produces the *rectified data representation* x_i. This is passed to the change detection phase implemented by the Adaptive Cusum algorithm, which consists of a standard deviation estimator that adaptively computes the Cusum threshold H^*, an EWMA filter to estimate the time series mean μ_i, and the Cusum algorithm itself which outputs the change detection events.

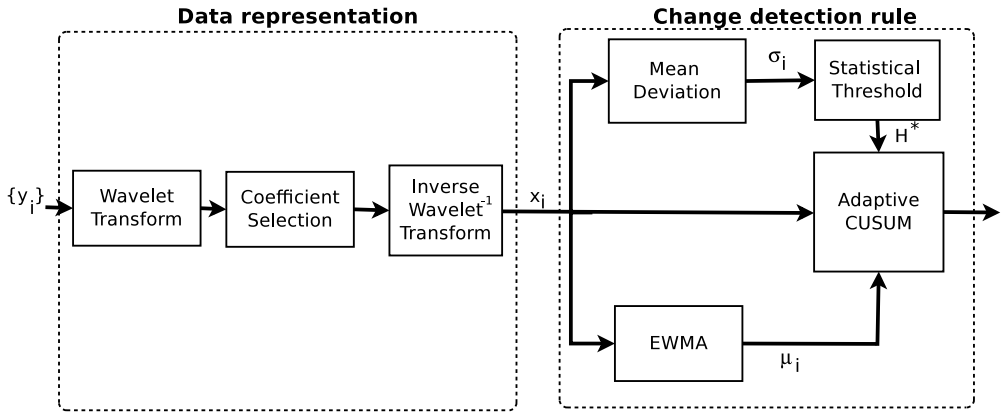


Fig. 7. Schematic view of the proposed methodology.

5. Other detection algorithms

The choice of the most appropriate model for state change detection depends on several factors, among which the application context and the statistical properties of the time series are the most important. These factors guide the choice of the detection rule and the data representation technique most appropriate to the state change detection model. As the focus of this paper is on state change detection algorithms working online in the context of Internet-based systems, many existing models cannot be considered for comparison because they can work just offline. Hence, as a term of comparison, we consider three popular classes of models that can be applied to online contexts: threshold-based model [20], Exponential Weighted Moving Average (EWMA) [6], and baseline Cusum [8].

The second important element is that the considered data streams are characterized by a high variability to the extent that, for a fair comparison, all considered algorithms work on a filtered data representation instead of raw data. To this purpose, we consider three rectification techniques: Exponential Weighted Moving Average (EWMA) [6], Kalman filter [12] and wavelet [17]. The combination of state change detection algorithms and filtering methods allows us to consider the following five detectors:

- EWMA$_n$-Threshold (Section 5.1).
- EWMA$_n$-EWMA (Section 5.2).
- Kalman-baseline Cusum (Section 5.3).
- EWMA$_n$-baseline Cusum (Section 4.1).
- Wavelet-baseline Cusum (Section 4.1).
- Wavelet-adaptive Cusum (Section 4.2).

5.1. EWMA$_n$-Threshold detector

Threshold-based models are commonly applied to Internet-based systems to support many real-time management algorithms [31–33]. The threshold-based detector uses a data representation x_i of the measurements that filters them through an exponential weighted moving average. It is defined as: $x_i = \lambda y_i + (1-\lambda)x_{i-1}$, where $0 \leq \lambda \leq 1$ (a value is $\lambda = \frac{2}{n+1}$ [6]) and the starting representative state value is the mean of n time series values, that is, $\mu_s = \sum_{i=1}^{n}(y_i)/n$. On the basis of the chosen number of n past values, we denote the data representation technique as EWMA$_n$. We consider a detection based on a single threshold [20]. The detection rule compares the deviation among the value of x_i and the estimated mean μ_s to a statistical threshold set equal to Δ, that is, the smallest shift to be detected. Therefore, the considered detection rule is:

$$\text{detection rule} = \begin{cases} \text{state change}, & \text{if } |\mu_s - x_i| \geq \Delta \\ \text{no state change}, & \text{otherwise.} \end{cases} \tag{22}$$

When a state change is detected, μ_s is updated to the current value of the data representation x_i. The performance of the threshold-based detectors depends on the choice of the threshold value. There is no theoretical support for it, and the right choice of the threshold value completely depends on the application context and the workload characteristics. The quality of the threshold-based detectors tends to worsen when working on time series with high variances, that are responsible for many false positive detections.

5.2. EWMA$_n$-EWMA detector

The Exponential Weighted Moving Average (EWMA) is largely used to detect state changes in control charts [6] that are tools to determine whether a process is in a stable statistical control. The data representation x_i is an exponential weighted moving average computed on n past values. This algorithm detects a relevant state change each time the shift among μ_s and x_i overcomes a threshold depending on Δ, on the standard deviation σ_y of the time series and on the cut-off frequency λ of the EWMA data representation. The detection rule is defined as:

$$\text{detection rule} = \begin{cases} \text{state change}, & \text{if} \quad |\mu_s - x_i| \geq M\sigma_y\sqrt{\dfrac{\lambda}{2-\lambda}} \\ \text{no state change}, & \text{otherwise} \end{cases} \tag{23}$$

where M is the length of the control limits and depends on the minimum shift Δ to be detected. μ_s is set to the current value of the data representation x_i each time the model detects a state change.

The EWMA detector is a good model when small shifts have to be detected [6]. In these conditions, the performance of the EWMA is similar to that of the baseline Cusum. Otherwise, when the state change consists of a large shift, the quality of the EWMA tends to decrease [6]. Moreover, the EWMA-based detectors are characterized by a tradeoff between the false positive detections and the number of false negative detections. A detector using a small set n of past values offers a more reactive data representation and tends to maximize the number of true positive detections at the cost of some false positive detections. Increasing the number n of considered past values, the data representation becomes smoother and should decrease the number of false positive detections at the cost of some false negatives. For this reason, in our context it is hard to find an n value able to provide reliable and efficient performance for time series with time-varying characteristics.

5.3. Kalman-baseline Cusum detector

State change detection models with the Kalman filter as data representation and the baseline Cusum as the detection rule are widely used in literature [7,34]. The traditional implementation of the Kalman filter requires the knowledge of the system state model, which is impossible to define in the Internet-based context due to its non-stationary and unpredictable behavior [9]. For this reason, we consider a simplified version of the Kalman filter, as presented in [12]. The considered data representation algorithm, namely BART, combines a version of the Kalman model for filtering the bandwidth measurements with a change detection model based on the Cusum test. The Kalman filter estimates the data representation x_i as follows:

$$x_i = x_{i-1} + G_i(y_i - Dx_{i-1}) \tag{24}$$

where G_i is the Kalman gain, y_i is the measured quantity and the D matrix represents its one step model. A Cusum based algorithm is then applied to x_i to update quickly the system state value when a state change is detected. This makes it feasible to overcome the tradeoffs regarding speed of adaptation to changes versus stable estimation. The BART algorithm can be directly applied in our contexts. We use the model parameter values that are tailored in [12].

The output of this algorithm is used as the data representation for the baseline Cusum detection rule. The quality of this detection model is conditioned by the ability of the BART algorithm to maintain a stable estimation of the state, through the choice of the statistical threshold H and of the slack value K that represents the minimum deviation that the detection rule of the baseline Cusum accounts for. The BART data representation tends to reduce significantly its quality in non-stationary conditions and when the time series variance increases because it updates continuously its state also during periods of stability. This tends to cause a high number of false detections.

6. Performance results

In this section, we present the main performance metrics (Section 6.1). Then, we evaluate the quality of the proposed online detection model in different time series conditions carried out by modifying several parameters of the original time series in terms of variance and perturbations (Section 6.2).

6.1. Performance metrics

The detection quality of the state change detection algorithms is evaluated in terms of the well known metrics *recall*, *precision* [35] and *mean delay* for detection [7].

We define *recall* as the fraction of detections that are relevant to the time series and that are successfully retrieved:

$$recall = \frac{TP}{TP + FN} \tag{25}$$

This measure can be considered as the probability that a change is successfully detected by the algorithm. To achieve a recall value equal to 1, the detection algorithm must signal all relevant changes. The value of the recall alone is insufficient,

Table 1
Recall - $\rho_y = 0$.

σ_y	EWMA$_5$ Threshold	EWMA$_5$ EWMA	EWMA$_{10}$ EWMA	EWMA$_5$ Baseline Cusum	Wavelet Baseline Cusum	Kalman Baseline Cusum	EWMA$_5$ Adaptive Cusum	Wavelet Adaptive Cusum
0.1	1	1	1	1	1	1	1	**1**
0.2	0.97	1	1	1	1	1	1	**1**
0.3	0.96	1	1	1	1	1	1	**1**
0.4	0.92	1	1	1	1	1	1	**1**
0.5	0.89	1	0.98	0.99	1	1	1	**1**
0.6	0.83	0.99	0.87	1	1	1	1	**1**
0.7	0.85	0.95	0.31	1	1	1	1	**1**
0.8	0.92	0.88	0.02	1	1	1	0.99	**1**
0.9	0.94	0.71	0.01	0.99	1	1	0.99	**1**
1	1	0.64	0.01	1	1	1	1	**1**

because it must be supported by some information related to the number of false detections. This is measured by the *precision*, that is the fraction of relevant detections:

$$precision = \frac{TP}{TP + FP} \tag{26}$$

where $(TP + FP)$ is the total number of detections. The precision gives information on the ability of a detection algorithm to limit false state change detections. A precision equal to 1 means that the algorithm detects only relevant changes, while low precision values are caused by a detection algorithm that signals many false state changes.

A tradeoff between recall and precision values exists. These two metrics can be combined into one measure, namely the $F - measure$, that gives a global estimation of the detection quality. The F-measure is the weighted harmonic mean of the precision and recall, that is,

$$F - measure = 2\frac{precision * recall}{precision + recall} \tag{27}$$

An F-measure value close to 1 denotes a good detection quality, while it is lower for detection algorithms affected by false positive and false negative detections.

The *mean delay* for detection is related to the ability of an algorithm to detect a state change when it actually occurs. It quantifies the time required for the detection of a new state through the distance between the sample at which the model signals a state change and the actual sample of change in the state representation, and computes the mean over all the state changes. For example, let us consider a time series with Y state changes. Let $[s_1, \ldots, s_Y]$ be the actual samples of change and $[\hat{s}_1, \ldots, \hat{s}_Y]$ the samples at which the model detects the changes. The mean delay for detection is defined as:

$$mean\ delay = \frac{\sum_{i=1}^{Y}(\hat{s}_i - s_i)}{Y} \tag{28}$$

Good detection algorithms should minimize the mean delay for detection.

6.2. Detection quality

As we are interested in online detections of changes in non-stationary, unpredictable time series arising from real application contexts, in this section we evaluate the detection quality of the proposed algorithm for a wide range of time series based on emulated profiles deriving from real measures. We consider time series with a relevant increment of their values followed by a proportional decrement as in [36]. To facilitate the analysis and algorithm comparisons, the profile is normalized so that state increases/decreases are denoted by a unit value and the model parameter values are set to provide the best results in every considered time series.

As expected, all detection algorithms tend to diminish their detection quality for increasing variability of the time series. It is important to apply the detection algorithms on time series characterized by different levels of variability. As described in [37,38], the most important statistical properties that characterize a time series are the standard deviation σ_y, and the data correlation index ρ_y. σ_y measures the dispersion of data while ρ_y measures the dependence of perturbations on the behavior of time series values. To this purpose we evaluate the performance metrics of the detection algorithms as a function of σ_y and for ρ_y set to 0. The recall and precision results for all considered algorithms and for several σ_y values are reported in Tables 1 and 2, respectively.

Table 1 shows that when the dispersion is low ($\sigma_y \leq 0.5$) all the considered algorithms achieve recall values close to 1. This means that the algorithms are able to detect all relevant state changes, with the exception of the threshold-based algorithm. When the dispersion increases ($\sigma_y > 0.5$), the algorithms using EWMA as a detection rule (i.e., EWMA$_5$-EWMA and EWMA$_{10}$-EWMA) worsen significantly, as can be seen from their recall values. They risk being completely unreliable

Table 2
Precision - $\rho_y = 0$.

σ_y	$EWMA_5$ Threshold	$EWMA_5$ EWMA	$EWMA_{10}$ EWMA	$EWMA_5$ Baseline Cusum	Wavelet Baseline Cusum	Kalman Baseline Cusum	$EWMA_5$ Adaptive Cusum	Wavelet Adaptive Cusum
0.1	1	0.23	0.31	1	1	1	1	**1**
0.2	0.33	0.43	0.50	0.99	1	1	1	**1**
0.3	0.13	0.50	1	0.95	1	1	1	**1**
0.4	0.08	0.97	1	0.89	1	0.99	1	**1**
0.5	0.06	0.98	1	0.77	0.99	0.98	0.99	**0.99**
0.6	0.05	0.81	1	0.65	0.90	0.87	0.92	**0.96**
0.7	0.05	0.7	1	0.50	0.85	0.72	0.87	**0.96**
0.8	0.05	0.61	1	0.37	0.80	0.58	0.73	**0.93**
0.9	0.04	0.53	1	0.29	0.73	0.49	0.64	**0.87**
1	0.04	0.52	1	0.25	0.67	0.39	0.55	**0.84**

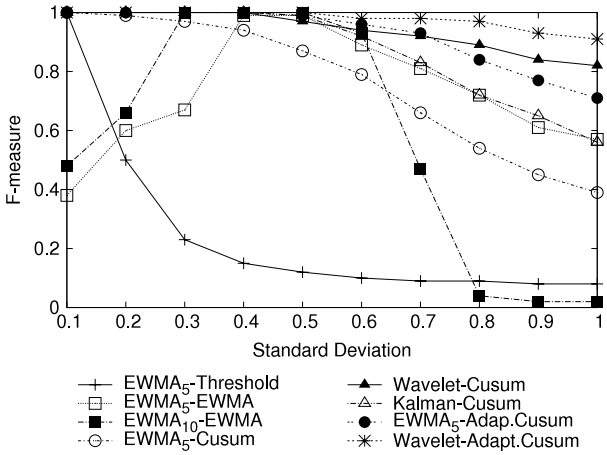


Fig. 8. F-measures.

in highly variable contexts, as they do not detect many state changes. Even in time series with an intense data dispersion, the threshold and Cusum-based algorithms have good recall values ($>$0.9), that is, they detect all relevant state changes. However, if we also consider the precision results, we notice in Table 2 that the threshold-based method is very sensitive to the data variations of the time series: although it detects all state changes, this is accompanied by a high number of false positives, as confirmed by the low precision values of the $EWMA_5$-Threshold. All Cusum-based algorithms are able to achieve good precision in time series characterized by a data dispersion $\sigma_y \leq 0.5$. In more variable contexts (e.g., $\sigma_y > 0.5$), only the proposed wavelet-Adaptive Cusum provides a high detection quality by achieving always a precision $\geq$0.84.

Fig. 8 reports the F-measure as a function of the standard deviation σ_y for all considered detection algorithms. This shows the combined effect of recall and precision. With the exception of algorithms based on the EWMA detection rule, the algorithm performance worsens for increasing values of σ_y. Fig. 8 shows that the wavelet-Adaptive Cusum algorithm achieves the best F-measure values for every σ_y. For example, when $\sigma_y = 1$, the F-measure of wavelet-Adaptive Cusum remains consistently above 0.9, even though EWMA Adaptive Cusum, the best existing online detection algorithm, has an F-measure of only 0.7. The threshold-based method is characterized by an exponential decay of the detection quality for increasing values of σ_y. This behavior reveals that a similar algorithm cannot be applied in non-stationary and non-deterministic contexts. For small standard deviations, the $EWMA_5$-EWMA and $EWMA_{10}$-EWMA improve the F-measure until $\sigma_y = 0.5$; beyond this, their F-measure decreases. This is due to their *inertia* limit that, together with the F-measure degradation for high σ_y values, highlights that the performance of the EWMA-based algorithms are unacceptable because they are too sensitive to the statistical characteristics of the time series and to the choice of algorithm parameters. Existing Cusum-based methods ($EWMA_5$-Cusum and Kalman–Cusum) are characterized by a small decay of the F-measure for low values of σ_y. On the other hand, the F-measure decreases faster when $\sigma_y > 0.5$. These results confirm that popular Cusum-based algorithms do not work well in online contexts related to system resource measures of Internet-based systems. Instead, combining the Cusum detection rule with a data representation based on the wavelet, it is possible to contain this limit as confirmed by the slow decay of the F-measure for high σ_y values.

We have to consider also the mean delay results, that are reported in Fig. 9 for all the considered algorithms. In Fig. 9, we plot the mean delay for detection for increasing values of σ_y. An expected trait of all the state change detection algorithms is the increase of the mean delay for detection, as a consequence of higher values of σ_y. All the algorithms but three are characterized by similar and low mean delay values. The $EWMA_{10}$-EWMA, $EWMA_5$-EWMA model and Kalman–Cusum, instead, are characterized by significant higher delays. In particular, the $EWMA_{10}$-EWMA is affected by a mean delay of

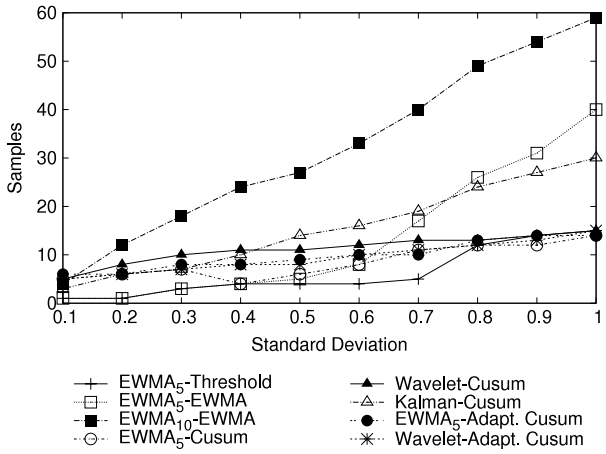


Fig. 9. Mean delay for detection.

60 samples in the worst case. This result shows that this algorithm is inadequate to support real-time management decisions. Moreover, it confirms that the choice of the data representation technique is crucial for online state detection models. In highly variable contexts, just the proposed wavelet-Adaptive Cusum provides an efficient tradeoff between the detection quality, expressed in terms of F-measure, and the delay.

Next, we examine the effects of the correlation ρ_y of the data component on the detection quality by considering different correlation indexes ($\rho_y = \{0, 0.1, 0.2, 0.3\}$) for two dispersion values ($\sigma_y = \{0.6, 0.9\}$). Fig. 10 reports the F-measure values of the four Cusum-based algorithms that in Fig. 8 show the best results (wavelet-Adaptive Cusum, $EWMA_5$-Adaptive Cusum, Kalman–Cusum and wavelet-Cusum) and of the two algorithms based on EWMA detection rule ($EWMA_5$-EWMA, $EWMA_{10}$-EWMA). The results confirm that the proposed wavelet-Adaptive Cusum algorithm improves the performance of all existing detectors for every correlation index and any σ_y characterizing the time series. For quite high values of noise dispersion ($\sigma_y = 0.6$), the wavelet-Adaptive Cusum algorithm always provides F-measures higher than 0.95. Its performance remains acceptable also when dealing with more variable time series, to the extent that in the most chaotic context of intense variance and strong correlation of the data component ($\sigma_y = 0.9$, $\rho_y = 0.3$) it improves by more than 50% the performance of the best existing algorithm wavelet-Cusum. This result confirms the importance of using an adaptive detection rule in non-stationary and highly variable contexts. A second important result is that the wavelet-Adaptive Cusum algorithm is less sensitive to the statistical characteristics of the time series with respect to any other detection algorithm. For a fixed value of σ_y, the performance of the proposed model degrades slowly with the increase the correlation index, thus demonstrating that the detection quality of the wavelet-Adaptive Cusum is less affected by ρ_y than all the other Cusum-based algorithms. This stiffness is quite useful in all real contexts characterized by highly variable, non-stationary and non-deterministic behaviors of the measured data.

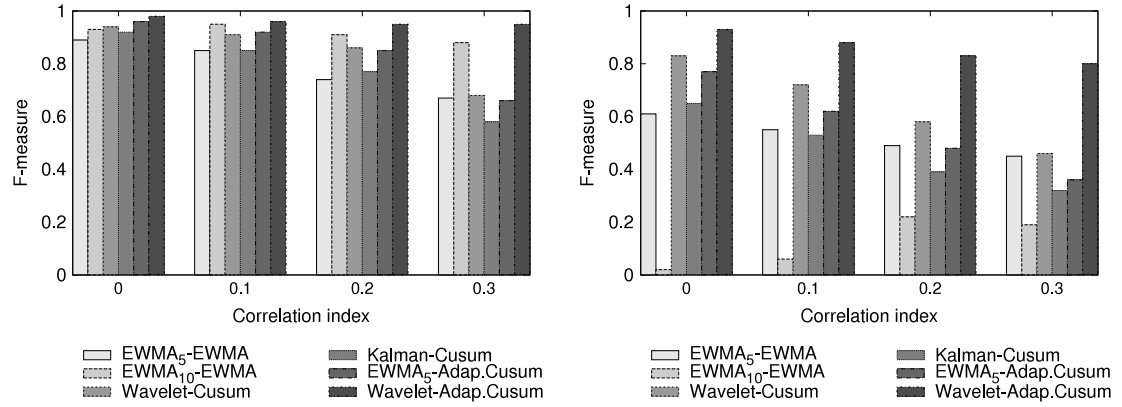
7. Experimental results

In this section, we apply the online detection models on measures related to system resources of real systems. In these experiments, since for real traces we do not actually have a *ground-truth* to refer to, we need a mean to define the representative state of time series and the occurrence of state changes. We adopted the following simple methodology[1], based on the *cluster analysis*. We start with an offline pre-processing of the monitored time series in which we remove all perturbations by means of a non-causal low-pass filter. We then apply cluster analysis to compute the representative states. We use a distance-based clustering based on the Quality Threshold clustering algorithm [39] and set a distance measure among clusters/states equal to Δ, that corresponds to the minimum relevance of the change that a model has to capture. For each state, we compute the mean value μ. The sequence of the mean values of the states $\{\mu\}$ and the respective samples $\{s\}$ to change to another state define the *representative states* of the time series.

In Fig. 11, we give an example of a real time series, representing the CPU utilization of a database server, and its representative states and samples of change. By applying cluster analysis with Δ equal to 0.2 to the time series of Fig. 11(a), we identify four representative states, defined by the following means values $\mu_1 = 0.50$, $\mu_2 = 0.75$, $\mu_3 = 0.52$ and $\mu_4 = 0.30$ and samples of change $s_1 = 50$, $s_2 = 100$ and $s_3 = 160$.

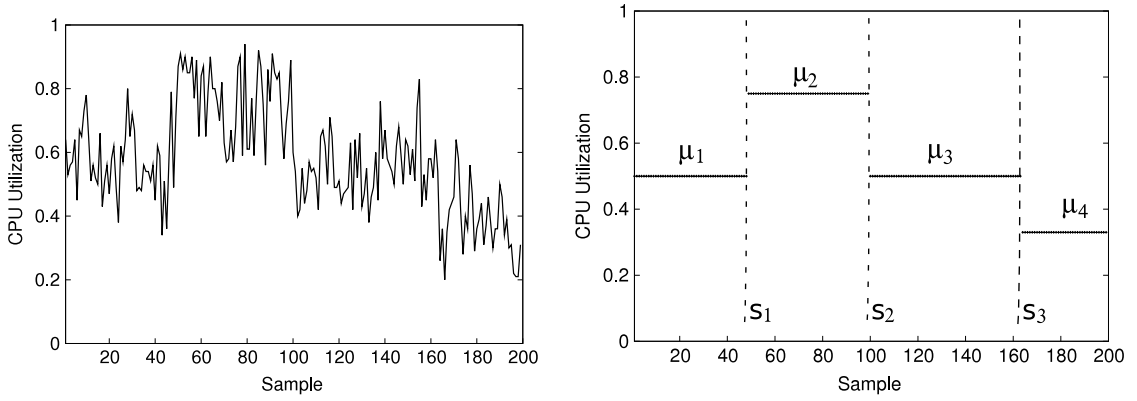
We evaluate the proposed state change detection algorithm on time series coming from server measures of an Internet-based system hosting Web sites with static, dynamic and secure content. For evaluation purposes, here we report the results

[1] We remark that in absence of a ground truth, any definition of representative state is arbitrary. Nevertheless, the proposed technique provided us *qualitatively* very good results.



(a) $\sigma y = 0.6$. (b) $\sigma y = 0.9$.

Fig. 10. F-measures.



(a) Time series. (b) Results of the quality threshold clustering.

Fig. 11. Representative state of a real time series.

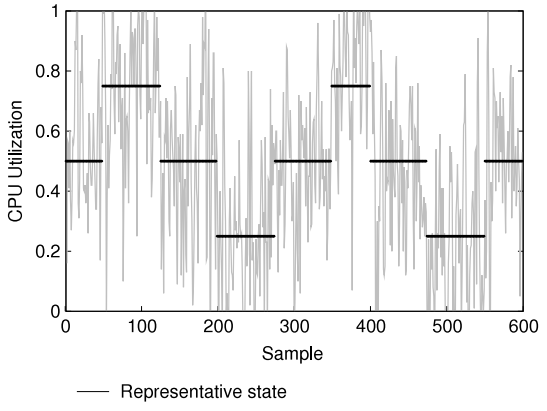
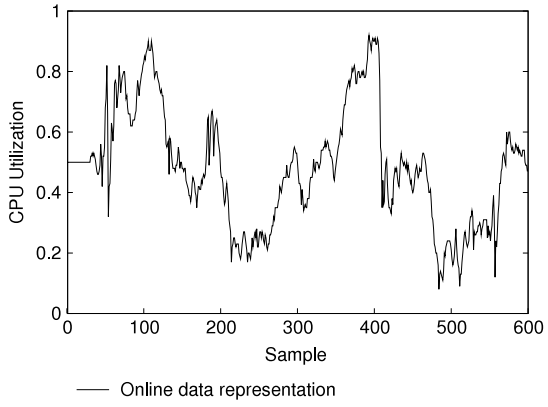


Fig. 12. CPU utilization of an Internet-based server.

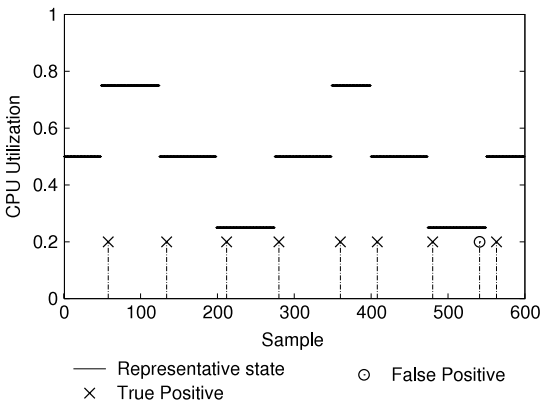
related to the measures shown in Fig. 12 referring to the CPU utilization of a machine hosting multiple guest servers. The considered measures are 1-minute averages of the CPU utilization on a single server.

We evaluate the representative state for the state change detection algorithms through the offline methodology described above that determines relevant state changes at samples 50, 125, 200, 275, 350, 400, 475, and 550, evidenced by the horizontal line in Fig. 12.

We report the results of the proposed wavelet-Adaptive Cusum and of the selected detection algorithms ($EWMA_5$-Cusum, Kalman–Cusum, $EWMA_5$-EWMA and $EWMA_{10}$-EWMA) described in Section 5.

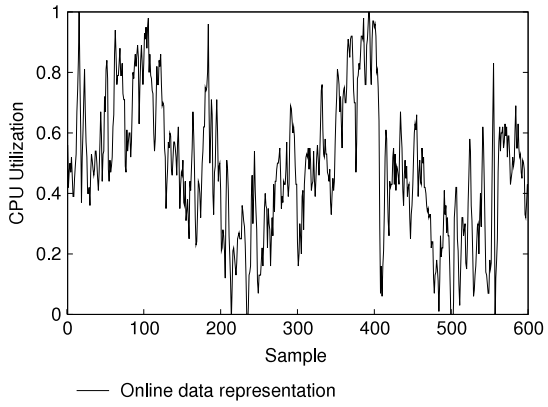


(a) Data representation.

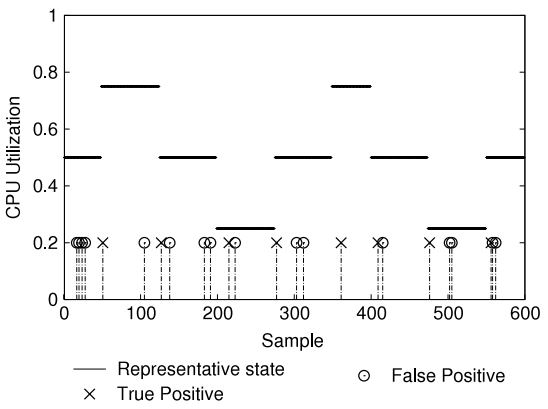


(b) State change detections.

Fig. 13. Wavelet-adaptive Cusum detector.

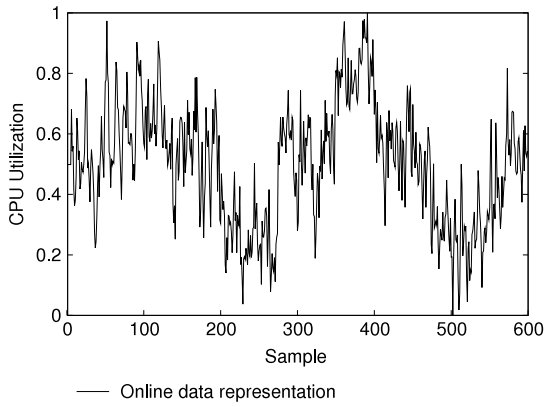


(a) Data representation.

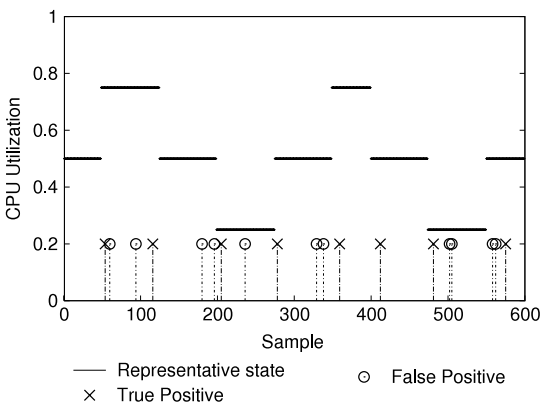


(b) State change detections.

Fig. 14. $EWMA_5$- detector.



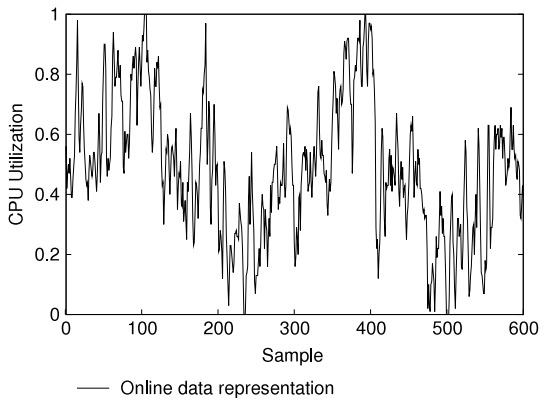
(a) Data representation.



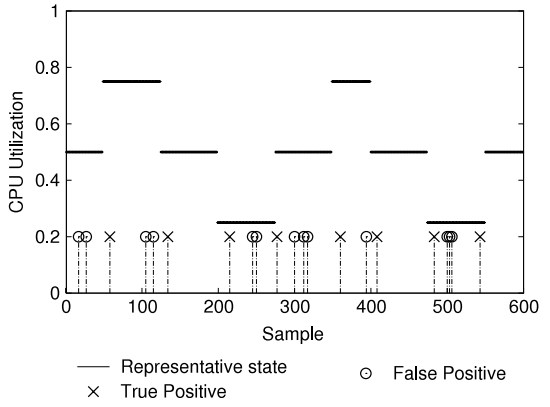
(b) State change detections.

Fig. 15. Kalman–Cusum detector.

Fig. 13(a), Fig. 14(a), Fig. 15(a), Fig. 16(a) and Fig. 17(a) show the curves of the online data representation generated by each of the five algorithms. The respective (b) figures on the right report the occurrence of false positive detections (vertical lines with a circle at the top) and true positive detections (vertical lines with a cross at the top). In this benchmark, we can appreciate that the wavelet-Adaptive Cusum algorithm is able to achieve a data representation more smoothed than the corresponding representations of the $EWMA_5$-baseline Cusum, the Kalman-baseline Cusum and the $EWMA_5$-EWMA and similar to the $EWMA_{10}$-EWMA.

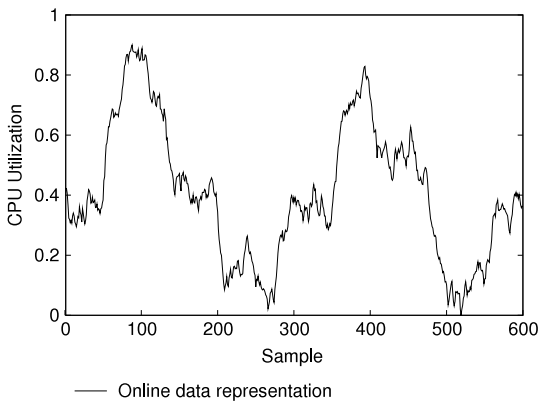


(a) Data representation.

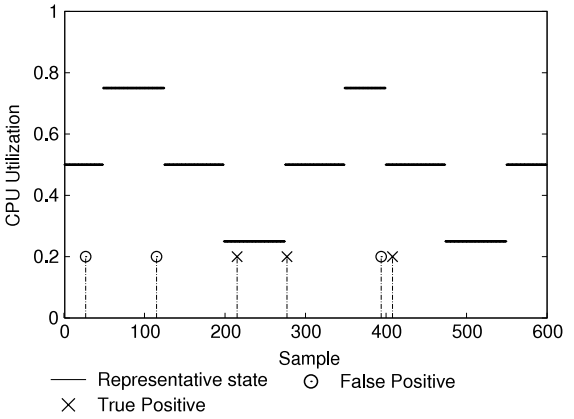


(b) State change detections.

Fig. 16. EWMA$_5$-EWMA detector.



(a) Data representation.



(b) State change detections.

Fig. 17. EWMA$_{10}$-EWMA detector.

The EWMA$_5$-Cusum, the Kalman–Cusum and the EWMA$_5$-EWMA shown in Fig. 14(b), Fig. 15(b) and Fig. 16(b), detect all state changes but their precision decreases due to a high number of false positive detections. This is a consequence of their inability to maintain a stable data representation when the time series are non-stationary and highly variable. A different behavior is shown by the EWMA$_{10}$-EWMA in Fig. 17(b): it misses more than 60% of state changes. This is due to the so-called *inertia limit* [6], that is, the inability to react quickly to time series changes when the size of the smallest shift to detect is significantly higher than time series variance. As a consequence, the inertia limit strongly affects the recall quality. These results show and confirm the tradeoff problem between the false positive detections and the number of false negative detections that characterizes the EWMA-based detection rule.

On the other hand, the wavelet-Adaptive Cusum algorithm in Fig. 13(b) exhibits a precise detection also in contexts characterized by multiple relevant changes and high variability. The proposed algorithm detects timely the state changes and it is affected by just one false detection at sample 540. Nevertheless, after the false state change detection, the wavelet-Adaptive Cusum algorithm is able to adapt immediately its data representation to the right stable state. This capacity of self-recovery is one of the most important properties of the proposed algorithm that allows us to achieve always the best results. We report a summary of the results shown in the previous figures in Table 3.

We now consider the problem of spiky time series. As an example, we consider the time series shown in Fig. 18, coming from the data streams related to the CPU utilization of a Web server. The measures exhibit a stable behavior without relevant state changes and are characterized by a low variance and by some spikes, such as at samples 170, 184, and 259. In Table 4, we report the number of false positives signaled by the considered state change detection models. A reliable detection model applied to this time series should not detect any relevant state change. However, in these conditions, only the wavelet-Adaptive Cusum avoids wrong detections. This result is the consequence of the usage of a reliable data representation rectified by spikes and perturbations and of the capacity of the proposed algorithm to preserve the stable state. On the other hand, the traditional state change detection models, such as the EWMA$_5$-Cusum, Kalman–Cusum and EWMA$_5$-EWMA,

Table 3
Summary of experimental results.

Model	TP	FP	FN
Wavelet-Adaptive Cusum	8	1	0
$EWMA_5$-Cusum	8	16	0
Kalman–Cusum	8	11	0
$EWMA_5$-EWMA	8	13	0
$EWMA_{10}$-EWMA	3	3	5

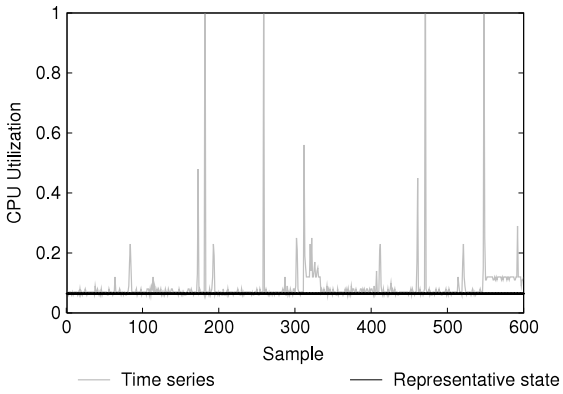


Fig. 18. Spike time series.

Table 4
Time series characterized by spikes.

Detector	False positive
Wavelet-Adaptive Cusum	0
$EWMA_5$-Cusum	10
Kalman-Cusum	4
$EWMA_5$-EWMA	8
$EWMA_{10}$-EWMA	2

detect many state changes in the occurrence of the spike values, as shown in Table 4. The $EWMA_{10}$-EWMA is able to limit the number of false detections to 2 because of the inertia limit.

8. Conclusions

Most real-time management decisions in large Internet-based infrastructures, from load balancing to access control to any autonomic-related control, rely on detection of relevant state changes of monitored system resources. In modern Internet-based systems characterized by virtual machines hosting several interactive Web-based applications we have observed that the data sets referring to system resource utilizations are characterized by non-deterministic and highly variable behavior. In this context, it is very tough to decide whether a significant state change has occurred and to signal it to the management framework. As existing online state change detectors do not work, we have proposed a new adaptive algorithm that is specifically tailored to be integrated with real-time management in Internet-based systems. The proposed algorithm combines for the first time an online version of the wavelet-based model to achieve a continuously adaptive representation of the data flowing from the system monitors with an online adaptive version of the well known Cusum statistical test as a detector. We demonstrate that the proposed algorithm is able to improve the performance of any existing state of the art detector applicable at runtime. All experiments carried out on real and modified time series demonstrate that the proposed solution is robust and effective: it signals just relevant state changes with very low numbers of false detections, and provides the precision of detection for any variability of the input data. The proposed algorithm is characterized by low computational complexity and can be applied to thousands of data flows. Larger sizes of input require hierarchical or parallel infrastructures for the evaluation.

Acknowledgments

We would like to express our gratitude to the many people who offered input on this work. Special and sincere thanks go to Michele Colajanni, Balachander Krishnamurthy and Giuseppe Serazzi for their valuable hints. We thank the anonymous reviewers of the InfQ 2010 Workshop for their helpful comments. We thank Mauro Andreolini and the companies, although they prefer not to appear, for providing data.

References

[1] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, Signal Processing, IEEE Transactions on 50 (2) (2002).
[2] C.K. Chui, G. Chen, Kalman filtering with real-time applications, Springer-Verlag, New York, Inc., 1987.
[3] R.Y. Rubinstein, D.P. Kroese, Simulation and the Monte Carlo Method, John Wiley & Sons, New York, 2007.
[4] F. Gustafsson, Adaptive Filtering and Change Detection, John Wiley and Sons, 2000.
[5] D.F. Allinger, S.K. Mitter, New results in innovations problem for nonlinear filtering, Stochastics 4 (1991).
[6] D.C. Montgomery, Introduction to Statistical Quality Control, John Wiley and Sons, 2008.
[7] M. Basseville, I. Nikiforov, Detection of Abrupt Changes: Theory and Application, Prentice-Hall, 1993.
[8] E.S. Page, Estimating the point of change in a continuous process, Biometrika 44 (1957).
[9] M. Andreolini, S. Casolari, M. Colajanni, Models and framework for supporting run-time decisions in web-based systems, ACM Tran. on the Web 2 (3) (2008).
[10] S. Casolari, F. LoPresti, T.S, Real-time models supporting management decisions in highly variable systems, in: Proc. of the 29th IEEE International Performance, Computing and Communications Conference, Dec. 2010.
[11] P. Dinda, D. O'Hallaron, Host load prediction using linear models, Cluster Computing 3 (4) (2000) 265–280.
[12] E. Hartikainen, S. Ekelin, Enhanced network-state estimation using change detection, in: Proc. of the 31st IEEE Conf. on Local Computer Networks, Nov. 2006.
[13] V. Chandola, A. Banerjee, V. Kumar, Anomaly Detection: Survey, ACM Computing Surveys, 2009.
[14] D. Kusic, J. Kephart, J. Hanson, N. Kandasamy, G. Jiang, Power and performance management of virtualized computing environments via lookahead control, in: Autonomic Computing, 2008. ICAC '08. International Conference on, 2008, pp. 3–12.
[15] N. Brenner, C. Rader, A new principle for fast fourier transformation, IEEE Acoustics, Speech & Signal Processing 24 (Mar) (1976) 264–266.
[16] S.G. Mallat, A theory of multiresolution signal decomposition: the wavelet decomposition, IEEE Trans. on Pattern Analysis and Machine Intelligence 11 (7) (1989).
[17] M.N. Nounou, B. Bakshi, On-line multiscale filtering of random and gross errors without process models, American Institute of Chemical Engineers Journal 45 (5) (1999).
[18] D.L. Donovo, I. Johnstone, G. Kerkyacharian, D. Picard, Wavelet shrinkage: Asymptotia? Journal of the Royal Statistical Society B 57 (2) (1995).
[19] D. Lu, P. Mausel, E. Brondizio, E. Moran, Change detection techniques, International Journal of Remote Sensing (2004).
[20] N.A. Macmillan, C.D. Creelman, Detection Theory: a User's Guide, Lawrence Erlbaum Associates, 2005.
[21] S. Casolari, M. Colajanni, F. LoPresti, Runtime state change detector of computer system resources under non stationary conditions, in: Proc. of 17th Int. Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Sept. 2009.
[22] A.S. Willsky, H.L. Jones, A generalized likelihood ratio approach to the detection and the estimation of jumps in linear systems, IEEE Transactions on Data and Knowledge Engineering 14 (2) (2002).
[23] H.W. Cain, R. Rajwar, M. Marden, M.H. Lipasti, An architectural evaluation of Java TPC-W, in: Proc. of the 7th Symposium on High Performance Computer Architecture, HPCA2001, Monterrey, ME, Jan. 2001.
[24] TPC-W transactional Web e-commerce benchmark, 2004.
[25] G. Moustakides, Optimal stopping times for detecting changes in distribution, The Annals of Statistics 14 (4) (1986).
[26] N. Vaswani, Additive change detection in nonlinear systems with unknown change parameters, Signal Processing, IEEE Transactions on 55 (3) (2007).
[27] D. Siegmund, Sequential analysis, in: Tests and Confidence Intervals, Springer, New York, 1985.
[28] M. Kendall, J. Ord, Time Series, Oxford University Press, 1990.
[29] V. Jaconson, Congestion avoidance and control, in: Proc. of SIGCOMM'88, vol. 21, Stanford, CA, Aug. 1988.
[30] C.K. Chui, An Introduction to Wavelets, Academic Press, 1992.
[31] G. Khanna, K. Beaty, G. Kar, A. Kochut, Application performance management in virtualized server environments, in: Proc. of Network Operations and Management Symp., 2006.
[32] N. Bobroff, A. Kochut, K. Beaty, Dynamic placement of virtual machines for managing sla violations, in: Proc. of the 10th IFIP/IEEE International Symp. on Integrated Network Management, 2007.
[33] T. Wood, P. Shenoy, A. Venkataramani, M. Yousif, Black-box and gray-box strategies for virtual machine migration, in: Proc. of the 4th USENIX Symp. on Networked Systems Design and Implementation, 2007.
[34] M. Severo, J. Gama, Change detection with kalman filter and cusum, in: Proc. of 9th Int. Conference on Discovery Science, Oct. 2006.
[35] D.L. Olson, D. Delen, Advanced Data Mining Techniques, Springer, 2008.
[36] M. Satyanarayanan, D. Narayanan, J. Tilton, J. Flinn, K. Walker, Agile application-aware adaptation for mobility, in: Proc. of the 16th ACM Intl. Symposium on Operating Systems Principles, Oct. 1997.
[37] M. Dobber, R. Vandet Mei, G. Koole, A prediction method for job runtimes in shared processors: Survey, statistical analysis and new avenues, Performance Evaluation (2007).
[38] B.L. Brockwell, R.A. Davis, Time Series: Theory and Methods, Springer-Verlag, 1987.
[39] L.J. Heyer, S. Kruglyak, S. Yooseph, Exploring expression data: identification and analysis of coexpressed genes, Genome Research (1999).

Sara Casolari is a researcher assistant at the Department of Information Engineering of the University of Modena and Reggio Emilia, Italy. She received her master degree (summa cum laude) and the Ph.D. in Computer Engineering form the University of Modena and Reggio Emilia in information engineering in 2004 and 2008, respectively.

Her research interests include stochastic models and performance evaluation of distributed systems, and modelling algorithms for supporting large systems and Internet-based application. She received a best paper award at the International Conference on Autonomic and Autonomous Systems (ICAS 2007) and at WWW/Internet 2010.

Stefania Tosi is a Ph.D. student in Information and Communication Technologies at the University of Modena and Reggio Emilia, Italy. She received her master degree (summa cum laude) in Computer Science from the same university in July 2010. Her research interests include performance evaluation of modern data centers and statistical models for data management. She received a best paper award at WWW/Internet 2010.

Francesco Lo Presti is an Associate Professor in the Department of Computer Science, Systems and Production of the University of Roma "Tor Vergata", Italy. He received the Laurea degree in electrical engineering and the Doctorate degree in computer science from the University of Rome "Tor Vergata" Rome, Italy, in 1993 and 1997, respectively. His research interests include measurements, modeling and performance evaluation of computer and communications networks. He has more than 50 publications in international conferences and journals. He has served as a program member of international conferences on networking and performance areas, and serves as reviewer for various international journals.