An Incremental Change Detection Test Based on Density Difference Estimation

Li Bu, Dongbin Zhao, *Senior Member, IEEE*, and Cesare Alippi, *Fellow, IEEE*

***Abstract*—We propose incremental least squares density difference (LSDD) change detection method, an incremental test to detect changes in stationarity based on the difference between the unknown prechange and the post-change probability density functions (pdfs). The method is computationally light and, hence, adequate to process continuous datastreams, as those emerging from the Internet of Things and the big data framework. The incremental change detection test operates on two nonoverlapping data windows to estimate the LSDD between the two pdfs. We construct a theoretical framework that shows how the distribution of LSDD values follows a linear combination of χ^2 distributions and provides thresholds to control false positive rates. The proposed test can operate online, with needed estimates and thresholds computed incrementally as fresh samples come. Comprehensive experiments validate the effectiveness of the test both in detecting abrupt and drift types of changes.**

***Index Terms*—Change detection, incremental computing, incremental least squares density difference change detection method (LSDD-Inc), probability density function (pdf)-free.**

I. Introduction

TIME invariance is a strong hypothesis to make when dealing with datastreams, no matter whether referring to learning problems [1]–[4] or control ones [5], [6]. In fact, in the long acquisition run, we cannot guarantee anymore that the interaction between sensors and the environment/system or the environment/system itself will not change [7], [8].

In order to cope with this very relevant issue, researchers have developed methods for an online detection of changes in stationarity, and methodologies to learn in such evolving environments. Most of existing research aims at detecting changes in stationarity in the datastreams by extracting features and inspecting associated statistics. Not rarely, features are extracted from two nonoverlapping data windows referring to prechange and possibly postchange conditions, respectively. The prechange window is composed of stationary samples x extracted according to some unknown, but fixed, probability density function (pdf) $p(x)$ to constitute the nominal reference set Z_p. The latter sliding window collects instead fresh samples as they come, extracted according to unknown pdf $q(x)$ to populate the test set Z_q. A change occurs in the sliding window when $q(x) \neq p(x)$. Following this comment, and given the fact we have only information about the prechange and post-change samples, a change occurs when "data in Z_q do not follow the distribution that generated Z_p according to a defined confidence level." The opposite holds.

Some change detection methods operate by comparing features extracted from the two windows, e.g., the sample mean or rank-based statistics, to detect changes in stationarity. In this direction, a change-point formulation is proposed in [9] to inspect changes affecting mean or variance in normally distributed samples. The method is then extended in [10] and [11] to deal with univariate non-Gaussian sequences. Changes are detected when designed statistics based on Mann–Whitney [12] and Lepage [13] tests exceed thresholds associated with predefined false positive (FP) rates. In order to deal with multivariate cases, [14] proposes a KNN-based test measuring the proportion of samples among k nearest neighbors that belong to a given window. It is shown that the derived statistics asymptotically satisfy a normal distribution, from which a threshold can be derived to meet a tolerated FP rate.

Only few papers attempt at directly comparing the known pdfs, e.g., with the KL-divergence or the Hellinger distance. The main restriction here is that reality is mostly pdf-free, in the sense that the distribution families are unknown. In order to handle this issue, researchers have found ways to estimate the pdfs directly from collected samples, with all associated limits, commonly by relying on histograms or kernel density estimation methods [15]. Most of the methods work incrementally to reduce the computational load associated with the integration of new samples in the change detection test. For instance, [16] suggests to extract frequency histograms from Z_p and Z_q and compare them according to the KL-divergence; a partition incremental discretization algorithm is applied to guarantee incremental computation. A different approach is proposed in [17], where frequency histograms are estimated with a kdp-tree computing the relative entropy: the tree is updated incrementally by adapting the corresponding nodes with new instances. A Gaussian mixture model (GMM) is considered in [18] to approximate the pdfs of the neighborhood

Manuscript received June 27, 2016; accepted March 6, 2017. Date of publication March 30, 2017; date of current version September 15, 2017. This work was supported by the National Natural Science Foundation of China under Grant 61573353, Grant 61533017, and Grant 61603382. This paper was recommended by Associate Editor F. Sun.

L. Bu and D. Zhao are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences and University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: bulipolly@gmail.com; dongbin.zhao@ia.ac.cn).

C. Alippi is with the Politecnico di Milano, 20133 Milano, Italy, and also with the Università della Svizzera italiana, 6904 Lugano, Switzerland (e-mail: cesare.alippi@polimi.it).

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/TSMC.2017.2682502

of each pixel in SAR images: the KL-divergence is then used to detect changes. However, the work in [19] shows how the Hellinger distance seems to be more attractive than the KL-divergence for change detection given its symmetry and boundedness properties. In [20], pdf $p(x)$ is approximated with GMM, while Lepage and one-sided t-tests are applied to monitor the changes in the log-likelihood of instances.

Last methods are based on a two-step procedure requiring at first to estimate the two pdfs and then evaluate their distance; variability associated with the finiteness of the data window and presence of noise in input data reduce the effectiveness of the methods. In order to mitigate this problem, some results present in the literature aim at directly measuring the density-ratio of the two distributions [21] or their density-difference [22] directly from available data windows. In our previous work [23], we investigated the performance and extended the least squares density difference (LSDD) method. A family of ensemble LSDD-based methods was also introduced in [24]. Even though effective, these last methods are computational intensive and lighter solutions must be proposed when computation is an issue.

In this paper, we investigate the LSDD method for change detection, shed light on some properties associated with the method and propose an incremental change detection test to reduce the computational request. The novel contributions reside as follows.

1) A theorem stating that the estimated LSDD values $\hat{D}^2_\lambda$ are distributed as a linear combination of noncentral chi-square distributions.
2) A theorem linking window size with FP and negative rates. As a consequence, the change detection test can adaptively enlarge the window size to improve detection performance without requesting any retraining phase.
3) Computationally light incremental algorithm for $\hat{D}^2_\lambda$.

The structure of this paper is as follows. Section II briefly recalls the LSDD method. Section III provides the main theoretical results and introduces the adaptive threshold mechanism. The detailed description of the incremental LSDD change detection method (LSDD-Inc) is given in Section IV. Finally, experiments showing the validity of the proposed change detection method are presented and commented in Section V.

II. LSDD Method

The LSDD is defined as the scalar

$$D^2(p,q) = \int (p(x) - q(x))^2 dx \tag{1}$$

where $x \in R^d$ is a real vector, and $p(x)$, $q(x)$ are two unknown pdfs. Instead of estimating $p(x)$ and $q(x)$, we directly estimate the difference $p(x) - q(x)$ with the Gaussian kernel model

$$g(x,\Theta) = \sum_{i=1}^{k} \theta_i \exp\left(-\frac{\|x - c_i\|_2^2}{2\sigma^2}\right) \tag{2}$$

where k is the number of kernel functions, c_i the ith kernel center, $\Theta = [\theta_1, \theta_2, \ldots, \theta_k]$ a parameter vector, and σ is a scale parameter.

The optimal parameter Θ is the one minimizing the loss

$$J(\Theta) = \int (g(x,\Theta) - (p(x) - q(x)))^2 dx + \lambda \Theta^T \Theta. \tag{3}$$

$\lambda > 0$ is an $L2$-regularizer controlling overfitting.

After some calculus, we obtain that

$$\begin{aligned} J(\Theta) &= \int g(x,\Theta)^2 dx - 2\int g(x,\Theta)(p(x) - q(x))dx \\ &\quad + \int (p(x) - q(x))^2 dx + \lambda \Theta^T \Theta \\ &= \Theta^T H \Theta - 2h^T \Theta + \int (p(x) - q(x))^2 dx + \lambda \Theta^T \Theta \end{aligned} \tag{4}$$

where H is a $k \times k$ matrix, and h a $k \times 1$ vector

$$\begin{aligned} H_{i,j} &= \int \exp\left(-\frac{||x - c_i||_2^2}{2\sigma^2}\right) \exp\left(-\frac{||x - c_j||_2^2}{2\sigma^2}\right) dx \\ &= \left(\pi\sigma^2\right)^{d/2} \exp\left(-\frac{||c_i - c_j||_2^2}{4\sigma^2}\right) \end{aligned} \tag{5}$$

$$\begin{aligned} h_i &= \int \exp\left(-\frac{||x - c_i||_2^2}{2\sigma^2}\right) p(x) dx \\ &\quad - \int \exp\left(-\frac{||x - c_i||_2^2}{2\sigma^2}\right) q(x) dx \end{aligned} \tag{6}$$

$i, j = 1, \ldots, k$. Defined $Z_p = \{x_{p,1}, \ldots, x_{p,n}\}$ as the data set drawn according to $p(x)$ and $Z_q = \{x_{q,1}, \ldots, x_{q,m}\}$ that from $q(x)$, Monte Carlo sampling provides estimates

$$\begin{aligned} \hat{h}_i &= \frac{1}{n}\sum_{l=1}^{n} \exp\left(-\frac{||x_{p,l} - c_i||_2^2}{2\sigma^2}\right) \\ &\quad - \frac{1}{m}\sum_{l=1}^{m} \exp\left(-\frac{||x_{q,l} - c_i||_2^2}{2\sigma^2}\right). \end{aligned} \tag{7}$$

Finally, $\hat{\Theta}$ is

$$\begin{aligned} \hat{\Theta} &= \arg\min_{\Theta} \left(\Theta^T H \Theta - 2\hat{h}^T \Theta + \lambda \Theta^T \Theta\right) \\ &= (H + \lambda I)^{-1} \hat{h}. \end{aligned} \tag{8}$$

By replacing $p(x) - q(x)$ with $g(x, \hat{\Theta})$, two equivalent expressions of the D^2-distance can be obtained

$$\hat{D}^2_1(p,q) = \int g\left(x, \hat{\Theta}\right)(p(x) - q(x))dx = \hat{h}^T \hat{\Theta} \tag{9}$$

$$\hat{D}^2_2(p,q) = \int (g(x,\hat{\Theta}))^2 dx = \hat{\Theta}^T H \hat{\Theta}. \tag{10}$$

In order to reduce the bias introduced by λ, we can write (details in Appendix A)

$$\begin{aligned} \hat{D}^2_\lambda(p,q) &= 2\hat{h}^T \hat{\Theta} - \hat{\Theta}^T H \hat{\Theta} \\ &= \hat{h}^T H_\lambda^{-1} \hat{h} \end{aligned} \tag{11}$$

where $H_\lambda^{-1} = 2(H + \lambda I)^{-1} - (H + \lambda I)^{-T} H (H + \lambda I)^{-1}$.

We comment that the higher $\hat{D}^2_\lambda$ the larger the discrepancy between $p(x)$ and $q(x)$.

III. Some Theoretical Results About LSDD

Estimate $\hat{D}_\lambda^2$ is a random variable depending on the particular realization of sets Z_p and Z_q as well as their cardinalities n and m, respectively. As such, in order to introduce a confidence level for FP, we need to determine the generating distribution and, then, the influence of the windows sizes n and m on the proposed method. Subsequent theorems address these aspects.

A. Distribution of $\hat{D}_\lambda^2$

Theorem 1: In stationary conditions, the distribution Π of $\hat{D}_\lambda^2$ is a linear combination of $k(k+1)$ noncentral chi-square distributions provided that kernel centers are given.

The proof derives from the central limit theorem and is given in Appendix B. The theorem also implicitly states that once n and m are fixed, so it is the resulting distribution of $\hat{D}_\lambda^2$.

Since the underlying distribution is now available, we can derive the threshold T_μ permitting to detect changes in stationarity at confidence level $1-\mu$ (the FP rate is hence set to μ). If the number of available samples is not enough to configure the parameters of the distributions, we can derive the required threshold, as suggested [17], as the $1-\mu$ percentile of the estimates so that

$$\Pr\left(\hat{D}_\lambda^2 > T_\mu\right) = \mu. \tag{12}$$

As a consequence, the hypothesis test behind the change detection test can be written as

$$H_0 : p(x) = q(x)$$
$$H_1 : p(x) \neq q(x).$$

Whenever pdf $q(x)$ differs from $p(x)$, values $\hat{D}_\lambda^2$ exceed T_μ with confidence level $1-\mu$, i.e., H_0 is rejected, and a change is detected. It has to be noted that the detected change occurs in the current sliding window Z_q, and we do not know the exact change location within the window. Other methods can be used to improve the location estimate, e.g., as proposed in [25].

In those cases where the training set is small and we cannot generate enough estimates for $\hat{D}_\lambda^2$ (say $[N_t/(n+m)] < 100$), we propose to use a bootstrap procedure [17], [23] to generate enough $\hat{D}_\lambda^2$ values to configure H_0. This procedure is appropriate since it is proved that bootstrap approximates the source distribution provided the pooling set is sufficiently informative [26].

In this paper, bootstrap operates as follows. At first, windows Z_p of size n, $Z_{p,i}, i = 1, \ldots, M$ and Z_q of size m, $Z_{q,i}, i = 1, \ldots, M$ are drawn from the stationary training set with replacement. The first M subsets are assumed to be generated from $p(x)$ and the second ones from $q(x)$. For the generic ith window couple $\{Z_{p,i}, Z_{q,i}\}$, the ith estimate $\hat{D}_\lambda^2$ is computed according to (11). The M couples are then representative of the situation in the stationary condition and used to configure test H_0. The needed threshold is then computed according to the predefined FP rate as shown above once a tolerated FP rate has been given.

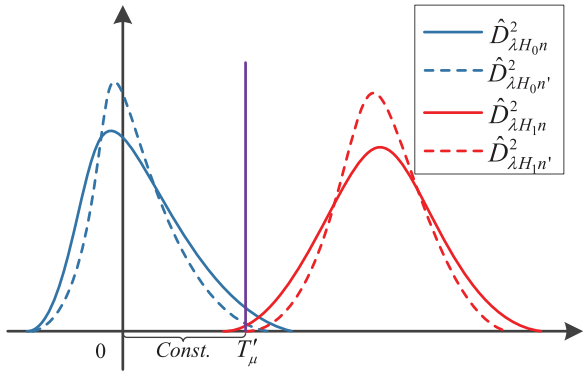

Fig. 1. Distribution of $\hat{D}_\lambda^2$ with different n. We centered the distributions and shifted threshold T_μ (to T'_μ) so that the new expectations of $\hat{D}_\lambda^2$ under H_0 are zero.

It should be emphasized that the proposed method only assumes the training set to be stationary to design the change detection test associated with H_0. This hypothesis is reasonable since time variance generally develops late in time, e.g., think of sensor aging.

B. Influence of the Window Size

The influence of the window size on the change detection test is presented as follows.

Theorem 2: The relationship between the window size and the LSDD statistics

1) The expectation of $\hat{D}_\lambda^2$ shows an inverse dependence in n and m

$$E\left(\hat{D}_\lambda^2\right) = f\left(\frac{1}{n}, \frac{1}{m}\right). \tag{13}$$

2) The difference of expectations $E_{H_1}(\hat{D}_\lambda^2) - E_{H_0}(\hat{D}_\lambda^2)$ inversely depends on m only

$$E_{H_1}\left(\hat{D}_\lambda^2(p,q)\right) - E_{H_0}\left(\hat{D}_\lambda^2(p,q)\right) = f_0\left(\frac{1}{m}\right). \tag{14}$$

3) The probability of an $\epsilon > 0$ deviation bound diminishes when n and m increase

$$\Pr\left(\left|\hat{D}_\lambda^2(p,q) - E\left(\hat{D}_\lambda^2(p,q)\right)\right| \geq \epsilon\right) \leq f_1\left(\frac{1}{n}, \frac{1}{m}\right). \tag{15}$$

The proof and functions f, f_0, and f_1 are given in Appendix C.

Theorem 2 indicates that the ϵ deviation bound decreases with n, whereas the difference between expectations does not, so that the overlap between the distributions Π_{H_0} and Π_{H_1} of $\hat{D}_\lambda^2$ diminishes.

The schematic of Fig. 1 shows how distributions change when the window size moves from n to $n' > n$. Define $T_{\mu'}$ to be the threshold associated with the larger size n', in turn associated with FP rate μ'. Since point 2) of Theorem 2 states that the difference of the expectations does not depend on n, we keep the distance between $E_{H_0 n'}(\hat{D}_\lambda^2)$ and $T_{\mu'}$ constant, that is

$$E_{H_0 n'}\left(\hat{D}_\lambda^2\right) - T_{\mu'} = E_{H_0 n}\left(\hat{D}_\lambda^2\right) - T_\mu. \tag{16}$$

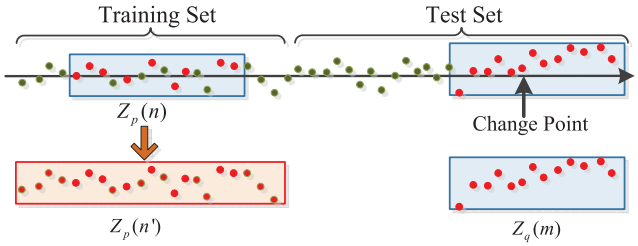

Fig. 2. Description of the enlarged reference set. The green points represent continuously generated samples and the red ones are samples included in subsets Z_p and Z_q, respectively. A change occurs in Z_q and the location is shown with a black arrow.

This comment leads to a major outcome since it permits us to both express the FP and false negative (FN) rates in terms of n'

$$\begin{aligned}&\text{FP rate}: \Pr\left(\hat{D}^2_{\lambda H_0 n} > T_\mu\right) = \mu > \Pr\left(\hat{D}^2_{\lambda H_0 n'} > T_{\mu'}\right) = \mu' \\ &\text{FN rate}: \Pr\left(\hat{D}^2_{\lambda H_1 n} < T_\mu\right) > \Pr\left(\hat{D}^2_{\lambda H_0 n'} < T_{\mu'}\right). \end{aligned} \tag{17}$$

In particular, (17) shows that a larger n' permits to achieve a better change detection performance by both lowering FP and FN rates. In addition, since the probability associated with the ϵ deviation bound also decreases with the increase of m (15), m also controls the FP rates so that a larger m reduces FPs. However, its influence on the FN rate is unknown at this stage of research.

C. Self-Adaptive Thresholds

During the operational phase, a single window of size n extracted from the training set represents a set of realizations following $p(x)$. At the same time m samples compose the sliding window associated with $q(x)$. A reservoir sampling method was proposed in [27] to mitigate the fact that a single window is considered, which updates Z_p to achieve lower FP rates [23]. Other approaches consider ensemble methods to include several reference windows to better represent $p(x)$ [24]. However, despite the fact we might consider those approaches, a natural question arises: "can we adapt the reference set Z_p to host more than n instances during the operational phase of the change detection test?" The answer to the question is clearly "yes," but it would *a priori* request a computationally expensive training phase. We then search for methods that can host more data with size n' ($n' > n$), and compute the new threshold $T_{\mu'}$ directly from the available estimate $\hat{D}^2_{\lambda H_0 n}$.

We propose in the sequel an adaptive mechanism for generating online the new thresholds as data come from the datastream. Initially, at training time, the change detection test undergoes a configuration phase where the limited size of the training set forces the designer to consider small values for n and m. Given n and m, threshold T_μ is derived accordingly. Then, during the operational phase, more stationary instances with size n' so that $n' > n$ are added to Z_p. We follow (16) to determine the new associated threshold $T_{\mu'}$, which permits to control the FP rates. An intuitive description of the enlarged reference set is shown in Fig. 2, where more stationary samples are included in Z_p.

Appendix C shows that in a stationary situation (H_0 holds), the expectation of $\hat{D}^2_\lambda$ with sizes n and m can be expressed as

$$E_{H_0 n}\left(\hat{D}^2_\lambda\right) = \left(\frac{1}{n} + \frac{1}{m}\right) C_1 \tag{18}$$

where C_1 is a constant depending on $p(x)$ and $q(x) = p(x)$. It can be proved that having n' and m, the new threshold becomes

$$\begin{aligned} T_{\mu'} &= E_{H_0 n'}\left(\hat{D}^2_\lambda\right) - \left(E_{H_0 n}\left(\hat{D}^2_\lambda\right) - T_\mu\right) \\ &= \left(\frac{\frac{1}{n'} + \frac{1}{m}}{\frac{1}{n} + \frac{1}{m}} - 1\right) E_{H_0 n}\left(\hat{D}^2_\lambda\right) + T_\mu \end{aligned} \tag{19}$$

where both the required expectation $E_{H_0 n}(\hat{D}^2_\lambda)$ and threshold T_μ are available.

Since Theorem 2 tells us that a larger m lowers the FP rate, it might be worth to enlarge the Z_q with size $m'(m' > m)$ and compute the associated threshold $T_{\mu''}$ with $\mu'' < \mu' < \mu$. We can write that

$$T_{\mu''} = \left(\frac{\frac{1}{n'} + \frac{1}{m'}}{\frac{1}{n} + \frac{1}{m}} - 1\right) E_{H_0 n}\left(\hat{D}^2_\lambda\right) + T_\mu. \tag{20}$$

This mechanism permits the method to operate online and, once a potential change is detected, to host more samples to decide whether confirm or reject the change in stationary hypothesis with better confidence by operating on n and m directly.

IV. LSDD-Inc: Incremental LSDD-Based Change Detection Test

An incremental computational approach is always appreciated when dealing with datastreams both to speed up the computation and relieve the storage needs. This section moves in this direction by proposing an incremental change detection test.

We comment that when the k kernel centers are given, both matrix H and H_λ^{-1} are fixed for a given λ. As a result, the online computation of (11) only requires $\hat{h}$ to be estimated, as Z_q updates with new instances. Value $\hat{h}$ at the $(i+m)$th sample can be expressed as

$$\begin{aligned} \hat{h}_{j(i)} &= C_2 - \frac{1}{m}\sum_{l=i+1}^{i+m} \exp\left(-\frac{||x_{q,l} - c_j||_2^2}{2\sigma^2}\right) \\ &= C_2 - f_2 \end{aligned} \tag{21}$$

where $C_2 = (1/n')\sum_{l=1}^{n'} \exp(-[(||x_{p,l} - c_j||_2^2)/2\sigma^2])$ represents a constant value associated with Z_p. Since we assume that the training set is stationary, we add the whole set into Z_p with $n' = N_t$ in this paper. It should be noted that Z_p can be enlarged further with $n' > N_t$ provided that new incoming are granted to be stationary.

When the test window Z_q slides and collects the $(i+m+1)$th instance, we have that

$$\begin{aligned} \hat{h}_{j(i+1)} &= C_2 - \frac{1}{m}\sum_{l=i+2}^{i+m+1} \exp\left(-\frac{||x_{q,l} - c_j||_2^2}{2\sigma^2}\right) \\ &= C_2 - f_2' \end{aligned}$$

Algorithm 1 LSDD-Inc

1: **Input**: Training set N_t, window sizes n and m, FP rate μ, number of resampled subsets $2M$;
Output: Change location.
2: Bootstrap $2M$ subsets from the training set, the first M subsets coming from distribution $p(x)$ with size n, and the remaining M ones from $q(x) = p(x)$ with size m;
3: Provide the M LSDD estimates according to (11);
4: Derive thresholds T_μ according to (12);
5: Take advantage of the whole training set so that $n' = N_t$; derive the new threshold $T_{\mu'}$ according to (19);
6: Prepare Z_q with m recently collected samples; $i = 1$;
7: Estimate C_2 and f_2 according to (21);
8: **while** (1) **do**
9: Incrementally estimate $f_2^{'}$ and update $\hat{h}$ according to (22);
10: Estimate the LSDD value $\hat{D}_\lambda^2$ with (11);
11: **if** $\hat{D}_\lambda^2 > T_{\mu'}$ **then**
12: Change in stationarity is detected at location i, with confidence $1 - \mu$;
13: *Break*;
14: **else**
15: Update sliding window Z_q;
16: $i = i + 1$;
17: **end if**
18: **end while**

$$f_2' = f_2 + \frac{1}{m}\left(\exp\left(-\frac{||x_{q,i+m+1} - c_j||_2^2}{2\sigma^2}\right) - \exp\left(-\frac{||x_{q,i+1} - c_j||_2^2}{2\sigma^2}\right)\right). \quad (22)$$

Within this incremental approach, at each time step, only two instances $x_{q,i+m+1}$ and $x_{q,i+1}$ need to be recomputed, and at most $(m + 1)$ samples need to be stored.

The final incremental change detection method is given in Algorithm 1. Since we deal with datasteams (i.e., data are generated continuously), the loop terminates (step 13) only when a change is detected. Reactions to the change, e.g., to update the application or retrain the detection method, can be considered.

V. Experiments

A. Datasets

To contrast the performances of the proposed incremental method LSDD-Inc with other change detection methods, seven applications are considered, including unidimensional and multidimensional ones. Since the exact change location in real applications is hardly available, most of the applications (D1–D6) are simulated. However, one real-world dataset (D7) is considered to test the effectiveness of the method in a real application.

1) Samples of application D1 follow a gaussian distribution $N(0, 0.5)$. The change induces a slow drift in the distribution toward distribution $N(0.5, 0.5)$.
2) Application D2 is inspired by a 10-D problem [28], whose instances satisfy a multivariate gaussian distribution. Means are fixed at $u_{1,i} = u_{2,i} = 0$; the covariance shifts from $\sigma_{1,ij(i=j)} = 0.5, \sigma_{1,ij(i\neq j)} = 0$ to $\sigma_{2,ij(i=j)} = 0.5, \sigma_{2,ij(i\neq j)} = 0.4$, $i, j = 1, \ldots, 10$.

TABLE I
Details of the Datasets

Dataset	*Type*	*Size*	*Dim*	*ChgType*	*AttrType*	*Multimodal*
D1	Syn	10000	1	drift	numerical	no
D2	Syn	10000	10	abrupt	numerical	no
D3	Syn	10000	2	abrupt	numerical	yes
D4	Syn	10000	2	drift	numerical	no
D5	Syn	10000	3	abrupt	numerical	no
D6	Syn	10000	3	abrupt	categorical	yes
D7	Real	9568	4	abrupt	numerical	no

3) Application D3 refers to a two-class rotating mixture of Gaussians application [29] with class centers shifting from $u_1 = [1/\sqrt{2}, 1/\sqrt{2}]$, $u_2 = [-1/\sqrt{2}, -1/\sqrt{2}]$ to $u_1 = [1/\sqrt{2}, -1/\sqrt{2}]$, $u_2 = [-1/\sqrt{2}, 1/\sqrt{2}]$. Covariance matrices are fixed at $\Sigma_1 = \Sigma_2 = [0.5, 0; 0, 0.5]$.
4) Application D4 refers to problem [30] with samples satisfying the restriction: $(x_1 - a)^2 + (x_2 - b)^2 \leq r^2$. Changes occur with the radius r slowly drifting from 0.2 to 0.3. $a = b = 0.5$; variables x_1 and x_2 are uniformly distributed in interval [0, 1].
5) Application D5 refers to a moving hyperplane problem [30] with $y \leq -a_0 + a_1x_1 + a_2x_2$. $a_1 = a_2 = 0.1$, a_0 shifts from -1 to -3.2; $x_{i(i=1,2)}, y$ are uniformly distributed in intervals [0, 1] and [0, 5], respectively.
6) Application D6 is the STAGGER problem [31] with categorical features. We transform this classification problem into a detection one by taking only one class of samples. Changes occur with concept1 shifting to concept2.
7) Application D7 is a real application with samples collected from a combined cycle power plant [32], [33], where hourly averaged temperature, ambient pressure, relative humidity and exhaust vacuum measurements are used to predict the net hourly electrical energy output. We normalize the dataset to interval $[-1, 1]$, and add a change by shifting the normalized temperature from x_1 to $-x_1$.

We summarize these datasets in Table I to show their different properties, where *Syn* is short for *synthetic*, *Dim* for *dimension*, *ChgType* for *change type*, and *AttrType* for *attribute type*.

B. Other Methods

Four methods are introduced for comparison, including our previous (LSDD-CDT test) [23], a statistical test (LogKStest) [20], an incrementally distance-based method (HDDDM) [19] and a hierarchical method (H-ICI) [34]. We point out that the last three methods are well established change detection methods, working either on pdfs or in 1-D applications.

We also propose *LSDD-Inc2*, an evolution of *LSDD-Inc*, that takes advantage of the fact FP rates reduce with larger m' as claimed by Theorem 2 (here we consider $m' = 2$ m). Needed thresholds follow (20).

In more details.

1) LogKStest: The test proposed in [20] detects changes by monitoring the log-likelihood of the pdf $p(x)$ of scalar x

$$L(x) = \log(p(x)).$$

By estimating $p(x)$ with a mixture of w Gaussians

$$\hat{L}(x) = -\frac{w}{2}\Big(\log\Big((2\pi)^d\Big)\det\big(\Sigma_{p,i^*}\big) + \big(x - u_{p,i^*}\big)'\Sigma_{p,i^*}^{-1}\big(x - u_{p,i^*}\big)\Big)$$

where i^* is the Gaussian of the mixture maximizing likelihood. A Kolmogorov–Smirnov test [20] is then used to test whether $\hat{L}(x)$ evaluated over the two windows follows the same pdf or not.

2) HDDDM: This method approximates $p(x)$ and $q(x)$ with histograms, and detects possible changes by inspecting their Hellinger distance [19]. When no changes are detected, the histogram of $p(x)$ is updated by adding samples of Z_q into Z_p incrementally, while the estimate of $q(x)$ is updated with the new acquired samples. The t-statistic is used to derive thresholds as suggested in [19]. Since the method is proposed for sequential batch learning, we take samples in one test window (nonoverlapped) as a batch, and use the same detection procedure as recommended in [19].

3) H-ICI: The *H-ICI* test [34] is a two-layered hierarchical CDT whose first level detects possible changes based on the intersection of confidence intervals (ICIs) rule [35], and the second one confirms changes with the Hotellings T-square statistic. The method can detect changes accurately with low FP rates, particularly in 1-D applications.

C. Experimental Setup

Given that only a finite training set is available and that the window sizes influence the distribution of $\hat{D}_\lambda^2$, n and m should be fixed when generating the bootstrap-based distribution approximating the real one. In addition, a smaller window size is always associated with a shorter execution time, which can be relevant in some applications. In this paper, we consider two configurations for training phase: $n = m = 100$ and $n = m = 200$.

Other needed parameters are chosen as follows. The size of training set N_t is 2000, changes in applications D1–D7 occur at sample 6001 and last to the end, and the number of taken bootstraps M is 2000. $k = n + m$, the kernel centers are randomly sampled from the training set and then fixed before training. The FP rate μ for most of the methods is set to 1%; μ_s, μ_w, and μ_c as requested by *LSDD-CDT* and corresponding to the three thresholds T_s, T_w, and T_c, respectively, are set to 10%, 2%, and 1%. Each experiment on each application is repeated 500 times.

The choice of the scaling parameter σ and the regularization parameter λ influences the accuracy of the density difference estimation method and the change detection performance. In this paper, σ is chosen as the median distance between instances in the training set $\sigma = \text{median}(||x_i - x_j||_2, 0 < i < j \leq N_t)$ [36], which is commonly used with a radial basis function kernel. With reference to Appendixes A–C, λ should be small to reduce the bias and is selected by controlling the relative difference (RD) between $\hat{D}_1^2$ and $\hat{D}_2^2$; RD is set to 0.2.

Since most of the applications follow Gaussians or uniform distributions, the maximum number W of Gaussians for *LogKStest* is set to $m/10$ based on experimental evidence. In this paper, $W = 10$ and 20 correspond to $m = 100$ and 200. *LogKStest*, *HDDDM*, and *H-ICI* keep the settings suggested in their relative manuscripts.

At last, we consider five indexes to evaluate the detection performance of the proposed *LSDD-Inc*.

1) *FP Rate [FP (%)]:* It represents the percentage that a test erroneously detected a change when no changes are present.
2) *FN Rate [FN (%)]:* It represents the percentage that an existing change is not detected.
3) *Accuracy [Acc (%)]:* It represents the percentage that changes are accurately detected when they occur; Acc = 1-FP-FN.
4) *Delay (Del in Samples):* It measures the promptness in change detection. A delay is recorded only when the change is accurately detected; both the mean and the standard deviation (in parentheses) are also provided.
5) *Computational Time [CT(s)]:* It measures the execution time needed to execute the test (reference platform: Intel Xeon X5650 at 2.66 GHz, 48 GB RAM, MATLAB R2011b). Results are averaged over 500 runs.

D. Abrupt Versus Drift Changes

The first experiment refers to an unidimensional gaussian distribution. The pdf in stationary conditions is $N(0, 0.5)$, and changes start at sample 6001 with the pdf shifting to $N(0.5, 0.5)$. The window sizes are $n = m = 100$. During the training phase, the first 2000 instances are used to derive the threshold T_μ associated with the predefined FP rate $\mu = 1\%$. During the test phase, the whole training set is used so that $n' = N_t$; the new threshold is $T_{\mu'}$ as given in (19).

Fig. 3 shows how the detection method operates in the case of an abrupt type of change [Fig. 3(a) and (c)] and a drift one [Fig. 3(b) and (d)].

The blue solid lines and the red dotted ones in Fig. 3(a) and (b) show the change location and the detected location, respectively. Changes can be detected immediately once the differences between Z_p and Z_q are significant, which explains why significant abrupt changes are detected earlier, whereas slow drifts introduce a larger detection latency.

E. FP and FN Rates

Here, we design two experiments applied to synthetic applications D1–D6 to verify how the real FP rates are aligned with the predefined, expected, ones; we then investigate FN rates.

The experiments follow the same training procedure described in Section IV. Then, the first experiment referring to FP rates continues to work on a stationary dataset, i.e., $p(x) = q(x)$. $2M_t$ subsets are randomly generated to provide M_t estimates, M_t of which populate Z_p (size n) and M_t populate the test set Z_q (size m). The second experiment about the FN rates

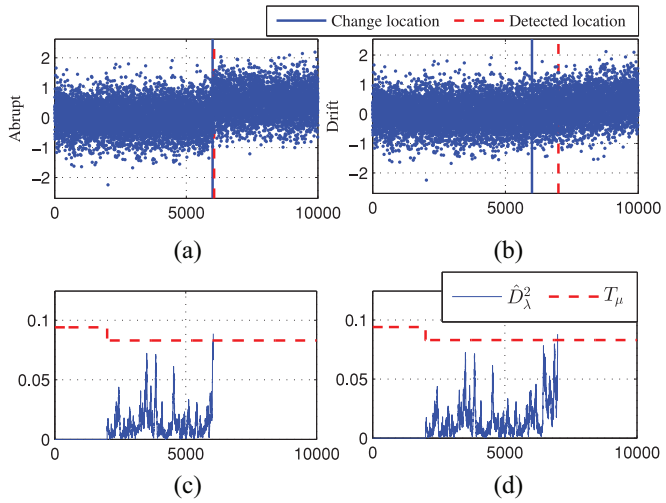

Fig. 3. Examples of detection performances with different change types: (a) abrupt and (b) drift changes. (c) and (d) Detection results, the blue line refers to the estimated LSDD values; and the red dotted line is the threshold. A change is detected once an LSDD value is above the threshold.

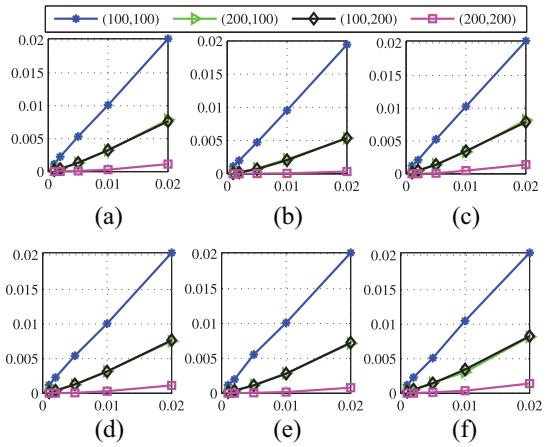

Fig. 4. Real FP rates versus the expected FP rates over different experiments. Lines represent the results with different (n, m) sizes. In the abscissas we have the expected FP value, on the ordinates the real computed FP rates. (a)–(f) D1–D6.

works on a nonstationary dataset where $p(x) \neq q(x)$. Finally, the real FP rate is computed on the first test, as the ratio of FPs on M_t tests, while the FN rate is experimentally assessed on the second experiment as the ratio of FNs.

In order to show the effectiveness of self-adaptive thresholds and the influence on FN rates, various combinations of sizes $n', m' \geq 100$ are considered during the test phase. The new threshold $T_{\mu''}$ is derived according to (20). In this experiment, M_t is set to 2000, the predefined FP rates belong to set $\{10\%, 2\%, 1\%, 0.2\%, 0.1\%\}$, and sizes n', m' to $\{100, 200\}$. Experiments are repeated 200 times to compute averaged FP and FN rates.

Results are shown in Figs. 4 and 5, respectively. Each subfigure in both figures shows the results on each application: (a)–(f) D1–D6 in Fig. 4; (a) D2, (b) D3, (c) D5, and (d) D6 in Fig. 5. In the two figures, the abscissas refer to the predefined FP rates, while ordinates refer to averaged FP

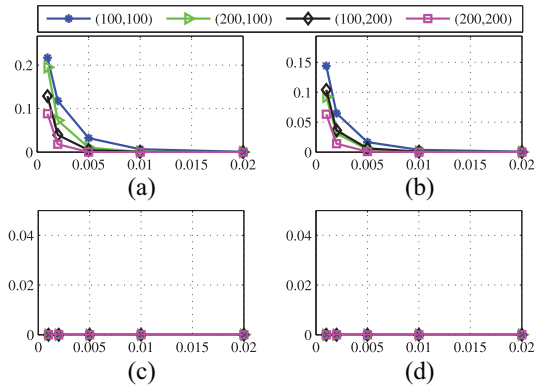

Fig. 5. Real FN rates under different predefined FP rates. Lines represent the results with different (n, m) sizes. In the abscissas we have the predefined FP rates, on the ordinates the real computed FN rates. (a) D2. (b) D3. (c) D5. (d) D6.

and FN rates, respectively. Lines represent results with different (n, m) sizes. Since drift changes in applications D1 and D4 are with different change rates, we do not record their FN rates.

As shown in Fig. 4, the real FP rates with $n, m = 100$ are close to the predefined ones, situation which indicates that the proposed method is effective in controlling the FP rates. Moreover, the FP rates with sizes $n(m) > 100$ are much smaller than the predefined values with $n = 100$, which yields to the expected conclusion that a larger window size helps to achieve lower FP rates. Results with cases $(100, 200)$ and $(200, 100)$ are similar and the corresponding lines overlap.

In Fig. 5(a) and (b), lower FP rates correspond to higher FN rates when the two distributions Π_{H_o} and Π_{H_1} overlap, which can be verified in Fig. 1. When changes are significant, the FN rates stay at zero as shown in Fig. 5(c) and (d).

F. Change Detection Performance

In this section, we compare the *LSDD-Inc* detection method with existing methods. Two different window sizes with $n = m = 100$ and $n = m = 200$ are applied to all methods, and during the test phases with *LSDD-Inc* and *LogKStest*, we update the reference Z_p with the whole training set, i.e., $n' = N_t$. For *LSDD-Inc2*, $n = m = 100$ during the training phase and $n = N_t, m' = 200$ in the test phase. The detection performance is shown in Table II; *ND* represents *not detected*.

In order to show performances of different methods, we apply statistical tests on the detection accuracy [Acc(%)].

For instance, the Friedman test verifies the significance of differences [37]. The test ranks the jth method for application Di r_i^j performance (1 is the best method). Ranking results are given in Table III, where average ranks are assigned in case of ties. The ranks for each method are averaged $R_j = (1/7)\sum r_i^j$ to compute the statistics

$$F_F = 11.51$$

where F_F satisfies an F-distribution with 9 and 54 degrees of freedom. However, the critical value of such a distribution $F(9, 54)$ with confidence level $\alpha = 0.05$ is 2.0585,

TABLE II
CHANGE DETECTION PERFORMANCE ON DIFFERENT APPLICATIONS

		LSDD-Inc		*LSDD-Inc2*	*LSDD-CDT*		*LogKStest*		*HDDDM*		*H-ICI*
Sizes	n, m	100	200	2×100	100	200	100	200	100	200	
D1	*FP(%)*	12	11.6	0.2	56.2	43.2	92.4	72.2	98	46.6	0
	FN(%)	0	0	0	0.2	0	0	0	0	0	0
	Acc(%)	88	88.4	99.8	43.6	56.8	7.6	27.8	2	53.4	**100**
	Del	1185.37 (372.56)	1096.1 (317.68)	1684.39 (299.78)	1526.99 (782.72)	1353.93 (523.05)	1121.11 (655.86)	1121.08 (718.53)	362.5 (199.55)	1325.52 (738.36)	1783.58 (241.13)
	CT(s)	2.31	3.86	2.65	20.42	28.09	5.46	11.84	0.01	0.0098	0.2
D2	*FP(%)*	5.6	5.6	0	70	60.2	94.4	82.6	69	1.2	0
	FN(%)	0	0	0	0	0	0	0	0	29.8	100
	Acc(%)	94.4	94.4	**100**	30	39.8	5.6	17.4	31	69	0
	Del	73.85 (11.22)	97.38 (11.68)	146.56 (12.75)	52.59 (13.24)	83.02 (14.68)	43.61 (14.23)	58.76 (15.82)	562.24 (436.99)	2127.7 (1011.3)	*ND*
	CT(s)	1.8	3.46	1.87	16.99	26.49	8.39	19.45	0.085	0.065	0.35
D3	*FP(%)*	16.6	11.2	0.2	55.8	40.4	93.2	72.6	97	26.6	0
	FN(%)	0	0	0	0	0	0	0	0	17	99.6
	Acc(%)	83.4	88.8	**99.8**	44.2	59.6	6.8	27.4	3	56.4	0.4
	Del	66.4 (9.04)	90.11 (13.59)	130.27 (12.25)	67.67 (13.02)	90.98 (19.78)	38.74 (9.88)	57.62 (23.9)	746.67 (445.4)	1785.8 (1037.47)	1690 (127.28)
	CT(s)	1.64	2.96	1.75	19.91	31.63	6.11	13.73	0.018	0.016	0.19
D4	*FP(%)*	13.8	10.4	0	52.2	41.4	94.6	87.8	95.4	22.6	0
	FN(%)	0	0	0	0	0	0	0	0	6.2	0
	Acc(%)	86.2	89.6	**100**	47.8	58.6	5.4	12.2	4.6	71.2	**100**
	Del	1587.87 (514.55)	1238.1 (347.63)	2324.7 (329.36)	1682.02 (751.34)	1321.43 (512.02)	353.19 (190.89)	405.7 (170.58)	1038.1 (629.66)	1580.81 (996.19)	1616.7 (221.1)
	CT(s)	1.96	3.2	2.29	17.23	27.83	6.12	13.14	0.018	0.016	0.29
D5	*FP(%)*	14.2	9.8	0	54.6	39.4	92.4	76.2	91	13.4	0
	FN(%)	0	0	0	0	0	0	0	0	0	0
	Acc(%)	85.8	90.2	**100**	45.4	60.6	7.6	23.8	9	86.6	**100**
	Del	29.47 (6.27)	42.25 (7.96)	57.61 (8.47)	28.86 (6.98)	41.15 (9.11)	20.82 (6.57)	31.61 (8.87)	100 (0)	200 (0)	146.56 (8.57)
	CT(s)	1.68	2.92	1.76	18.65	30.73	6.72	13.92	0.026	0.017	0.24
D6	*FP(%)*	17.2	12	0.4	54.2	41.8	-	-	98	66.4	100
	FN(%)	0	0	0	0	0	-	-	0	0	0
	Acc(%)	82.8	88	**99.6**	45.8	58.2	-	-	2	33.6	0
	Del	49.11 (4.81)	68.8 (6.89)	96.66 (6.29)	29.09 (4.08)	40.92 (5.7)	- -	- -	100 (0)	200 (0)	*ND*
	CT(s)	1.57	2.76	1.68	9.9	17.5	-	-	0.022	0.018	0.056
D7	*FP(%)*	99.2	65.6	0	67.6	55.8	81	78.4	100	0	0
	FN(%)	0	0	0	0	0	0	0	0	0	0
	Acc(%)	0.8	34.4	**100**	32.4	44.2	19	21.6	0	**100**	**100**
	Del	75.25 (0.5)	80.99 (1.87)	123.58 (9.02)	52.79 (7.01)	72.34 (9.28)	19.48 (1.57)	19.75 (3.43)	*ND*	3000 (0)	3568 (0)
	CT(s)	0.78	2.46	1.78	17.23	29.02	7.85	16.26	0.027	0.032	0.21

which is much smaller than F_F. Therefore, the null hypothesis is rejected, which means these detection accuracies are significantly different.

Then, we considered the Nemenyi test. The test investigates the differences between averaged ranks and the critical difference (CD)

$$\text{CD} = 5.1205.$$

However, most of the differences between $R_j (j = 1, \ldots, 10)$ are smaller than CD, which rejects the hypothesis about significant difference. Here, we can claim that *LSDD-Inc2* significantly outperforms *LogKStest*, whereas no such conclusion can be confirmed for tests *LSDD-Inc2* and *LSDD-CDT*.

We further conducted a pairwise comparison with the Wilcoxon Signed-Ranks Test. The method ranks the differences in accuracy of two methods for each dataset, and computes a statistic z, where the null hypothesis is rejected if z is smaller than -1.96 at confidence 0.95 [37]. It should be noted that for fair comparison, only different methods with the same window sizes or the same methods with different sizes are tested in pairs. Those above average rank 5.5, are removed because of their poor performance.

Results are shown in Table IV, where 1 indicates the rejection of the null hypothesis so that the compared methods are significantly different, 0 means nonrejection and * says that no comparison can be carried out. We can conclude the following.

1) *H-ICI* shows no significant detection performance differences when compared with other methods.
2) *LSDD-Inc2* is significantly better than other methods except *H-ICI*.

TABLE III
COMPARISON ON DETECTION ACCURACY: FRIEDMAN TEST

r	*LSDD-Inc*		*LSDD-Inc2*	*LSDD-CDT*		*LogKStest*		*HDDDM*		*H-ICI*
n, m	100	200	2×100	100	200	100	200	100	200	
D1	4	3	2	7	5	9	8	10	6	1
D2	2.5	2.5	1	7	5	9	8	6	4	10
D3	3	2	1	6	4	8	7	9	5	10
D4	4	3	1.5	7	6	9	8	10	5	1.5
D5	5	3	1.5	7	6	10	8	9	4	1.5
D6	3	2	1	5	4	9	9	7	6	9
D7	9	5	2	6	4	8	7	10	2	2
R	*4.36*	***2.93***	***1.43***	*6.43*	*4.86*	*8.86*	*7.86*	*8.71*	*4.58*	*5*

TABLE IV
PAIRWISE COMPARISON: WILCOXON SIGNED-RANKS TEST

		Inc.[1]		*Inc2.*[2]	*LSDD-CDT*	*HDDDM*	*H-ICI*
		100	200	2×100	200	200	
Inc.	100	*	1	1	*	*	0
	200	1	*	1	1	0	0
Inc2.		1	1	*	1	1	0
LSDD-CDT		*	1	1	*	0	0
HDDDM		*	0	1	0	*	0
H-ICI		0	0	0	0	0	*

1.*Inc.* is short for *LSDD-Inc*; 2.*Inc2.* is short for *LSDD-Inc2*.

3) Methods with larger window sizes, i.e., $n = m = 200$, outperform those with smaller ones, as expected.
4) Given the same window sizes, *LSDD-Inc* works significantly better than *LSDD-CDT* and *LogKStest*, but only slightly better than *HDDDM*.

It can be concluded that *LSDD-Inc* can detect changes more accurately than *H-ICI* and other methods when applying the same window sizes, and enlarging the test set Z_q will improve the detection performance with smaller FP rates, like *LSDD-Inc2*.

LogKStest shows the worst performance with the highest FP rates, although it reports a contained latency. Moreover, it can not deal with applications with categorical features, i.e., discrete values, because the estimation of pdfs with GMM fails to converge. In other cases with continuous pdfs, increasing sizes n, m will reduce the FP rates but cause larger detection delays.

HDDDM demonstrates similar performance as *LogKStest* especially with small window sizes, whereas it does not limit the attribute types. With the increase of n and m, the FP rates decrease, while the FN rates and detection delays unexpectedly increase. This mainly results from the inappropriate updating of the reference set. The data set has to be large enough to perceive the difference. In this case, if changes are not detected timely, nonstationary instances in Z_q will be removed into Z_p which may change the underlying pdf of the reference set.

H-ICI has the highest accuracy in some applications, but it fails to detect changes in multidimensional applications (D2-3) and application with categorical features (D6).

The *LSDD-CDT* presents acceptable results. However, since the reference set Z_p is almost fixed with only n samples, the FP rate appears to be high. Furthermore, the execution time is high because of the exhaustive computation of deriving $\hat{\Theta}$ and $\hat{h}$ with $(n + m)$ samples each time.

LSDD-Inc shows excellent performance with accurate detection in most applications. The detection delay is smaller than other methods when dealing with abrupt changes, and comparable when detecting drift ones. It also shows that with the increase of window sizes, the delay decreases in the drift cases, whereas it increases in the abrupt ones. This happens since in the drift case, more nonstationary samples in Z_q help to reveal the differences between Z_p and Z_q earlier. However, in the abrupt case, a larger window includes more stationary samples which lowers the differences. *LSDD-Inc* fails in application D7 where a small fluctuation occurs before the artificial change. The problem can be solved by considering a larger window size. In addition, thanks to the incremental computation, the execution time reduces compared to *LSDD-CDT*.

LSDD-Inc2 provides the highest accuracy. Thresholds $T_{\mu''}$ with $n' = N_t$, $m' = 2m$ contribute to reduce the FP rates as analyzed in Section III. FN rates are 0 in all the applications, since distributions Π_{H_0} and Π_{H_1} of estimates $\hat{D}^2_\lambda$ weakly overlap.

Even if *H-ICI* works perfectly in 1-D applications with continuous data, *LSDD-Inc* and *LSDD-Inc2* tests outperform other methods.

VI. CONCLUSION

In this paper, we propose an incremental change detection algorithm based on the LSDD method (LSDD-Inc). We prove that in stationary conditions, the estimate $\hat{D}^2_\lambda$ with fixed kernel centers is distributed as a linear combination of $k(k + 1)$ nonindependent noncentral Chi-square distributions. We provide a theoretical bound between the window size and FP rate, which permits the test to adapt the window size to improve detection performance without the need to retrain. During the training phase, a bootstrap-based distribution is considered to approximate the real one, and thresholds are derived according to the desired FP rates. When the window sizes increase, new thresholds can be determined directly from already available estimates. For online detection, we also estimate $\hat{D}^2_\lambda$ incrementally.

Comprehensive experiments show that the proposed method *LSDD-Inc* provides good performances in terms of promptness and accuracy.

APPENDIX A
CHOICE OF THE REGULARIZATION PARAMETER λ

Properties of Matrix H

As shown in (5), H is a real symmetric matrix. As such:

1) H can be decomposed as $H = V\Sigma V^T$, where V is an orthogonal matrix and Σ is a diagonal one;
2) all the elements on the diagonal $H_{j,j} (j = 1, \ldots, k)$ are equal to $(\pi\sigma^2)^{d/2}$. Furthermore, $H_{j,j} = \max(H) > \min(H) > 0$, where $\max(H)$ and $\min(H)$ describe the maximum and minimum values of elements in H, respectively.

Given any nonzero vector $Z = \{z_j, j = 1, \ldots, k\}$

$$\begin{aligned} Z^T H Z &= \sum_{i=1}^{k}\sum_{j=1}^{k} z_i z_j H_{i,j} \\ &> \min(H)\sum_{i=1}^{k}\sum_{j=1}^{k} z_i z_j \\ &= \min(H)\left(\sum_{i=1}^{k} z_i\right)^2 > 0 \end{aligned}$$

and matrix H is positive definite.

Write the diagonal matrix Σ as

$$\Sigma = \begin{pmatrix} r_1 & & & \\ & r_2 & & \\ & & \cdots & \\ & & & r_k \end{pmatrix}$$

where without loss of generality $r_1 > r_2 > \cdots > r_k > 0$. We have that $r_1 r_2 \cdots r_k = |H| < (\pi\sigma^2)^{kd/2}$, and $\sum_{i=1}^{k} r_i = k$.

Two Equivalent Expressions

From (9) and (10), the two equivalent expressions can be expressed as

$$\begin{aligned} \hat{D}_1^2(p,q) &= \hat{h}^T\hat{\Theta} = \hat{h}^T(H+\lambda I)^{-1}\hat{h} \\ \hat{D}_2^2(p,q) &= \hat{\Theta}^T H\hat{\Theta} = \hat{h}^T(H+\lambda I)^{-T}H(H+\lambda I)^{-1}\hat{h}. \end{aligned}$$

Since $(H+\lambda I)$ could be decomposed as $V\Sigma_\lambda V^T$ with $\Sigma_\lambda = \Sigma + \lambda I$, we transform the two expressions as

$$\begin{aligned} \hat{D}_1^2(p,q) &= \hat{h}^T V\Sigma_\lambda^{-1}V^T\hat{h} \\ \hat{D}_2^2(p,q) &= \hat{h}^T V\Sigma_\lambda^{-1}\Sigma\Sigma_\lambda^{-1}V^T\hat{h} \end{aligned}$$

where

$$\Sigma_\lambda^{-1} = \begin{pmatrix} \frac{1}{r_1+\lambda} & & & \\ & \frac{1}{r_2+\lambda} & & \\ & & \cdots & \\ & & & \frac{1}{r_k+\lambda} \end{pmatrix}$$

$$\Sigma_\lambda^{-1}\Sigma\Sigma_\lambda^{-1} = \begin{pmatrix} \frac{r_1}{(r_1+\lambda)^2} & & & \\ & \frac{r_2}{(r_2+\lambda)^2} & & \\ & & \cdots & \\ & & & \frac{r_k}{(r_k+\lambda)^2} \end{pmatrix}.$$

When $\lambda > 0$, $(1/r_i) > [1/(r_i+\lambda)] > [r_i/((r_i+\lambda)^2)] > 0, i = 1, \ldots, k$, where $(1/r_i)$ is the ith eigenvalue of H^{-1}. We can therefore conclude that $\hat{D}_1^2(p,q) > \hat{D}_2^2(p,q) > 0$.

Choice of Parameter λ

$\hat{D}_1^2(p,q)$ and $\hat{D}_2^2(p,q)$ can be weighted with parameter a to reduce the bias introduced by λ as

$$\hat{D}_\lambda^2(p,q) = a\hat{h}^T\hat{\Theta} + (1-a)\hat{\Theta}^T H\hat{\Theta}.$$

We have that

$$\hat{D}_\lambda^2(p,q) = \hat{h}^T V\begin{pmatrix} \frac{1}{\hat{r}_1} & & & \\ & \frac{1}{\hat{r}_2} & & \\ & & \cdots & \\ & & & \frac{1}{\hat{r}_k} \end{pmatrix}V^T\hat{h} \tag{23}$$

where $(1/\hat{r}_i) = [(r_i + a\lambda)/((r_i+\lambda)^2)]$ and its derivative with respect to λ is

$$\frac{d\left(\frac{1}{\hat{r}_i}\right)}{d\lambda} = \frac{(a-2)r_i - a\lambda}{(r_i+\lambda)^3}. \tag{24}$$

In addition, the ratio between the original eigenvalue $(1/r_i)$ and the new one is

$$\frac{\frac{1}{r_i}}{\frac{1}{\hat{r}_i}} = \frac{1}{r_i}\times\frac{(r_i+\lambda)^2}{r_i+a\lambda} = 1 + \frac{(2-a)r_i\lambda + \lambda^2}{r_i(r_i+a\lambda)}. \tag{25}$$

From (24), when $0 \le a \le 2$, the derivative is smaller than 0, which indicates the decreasing property of the new eigenvalues. The right-hand side of (25) is greater than 1 so that $(1/r_i) > (1/\hat{r}_i)$. Thus, with the decrease of λ, the value of $(1/\hat{r}_i)$ increases and approaches $(1/r_i)$, i.e., the smaller λ the smaller the bias.

By expanding $(1/\hat{r}_i)$ with Taylor

$$\begin{aligned} \frac{1}{\hat{r}_i} = \frac{1}{r_i} + \frac{(a-2)\lambda}{{r_i}^2} - \frac{(2a-3)\lambda^2}{{r_i}^3} \\ + \cdots + (-1)^{n+1}\frac{(na-(n+1))\lambda^n}{{r_i}^{n+1}} + R_n(\lambda) \end{aligned} \tag{26}$$

when $0 < \lambda < 1$, the setting of $a = 2$ could eliminate the influence brought by the low-order terms of λ. Finally, we have

$$\hat{D}_\lambda^2(p,q) = 2\hat{h}^T\hat{\Theta} - \hat{\Theta}^T H\hat{\Theta}$$

and $(1/\hat{r}_i) = [(r_i+2\lambda)/((r_i+\lambda)^2)]$ where $(1/\hat{r}_1) < (1/\hat{r}_2) < \cdots < (1/\hat{r}_k) < (2/\lambda)$.

On the other hand, λ is required to control overfitting which is essential to avoid the singularity of H. A method to control the RD between $\hat{D}_1^2$ and $\hat{D}_2^2$ has been proposed in [23]

$$\mathrm{RD} = \frac{\hat{h}^T\hat{\Theta} - \hat{\Theta}^T H\hat{\Theta}}{\hat{h}^T\hat{\Theta}} = 1 - \frac{\hat{\Theta}^T H\hat{\Theta}}{\hat{h}^T\hat{\Theta}}$$

that the largest λ ($\lambda < 1$) with corresponding RD smaller than a given constant is selected.

APPENDIX B
PROOF OF THEOREM 1

$\hat{D}_\lambda^2(p,q)$ can be represented as $\hat{D}_\lambda^2(p,q) = \hat{h}^T H_\lambda^{-1}\hat{h} = \sum_{i=1}^{k}\sum_{j=1}^{k} H_{\lambda(i,j)}^{-1}\hat{h}_i\hat{h}_j$ with $H_\lambda^{-1} = V(2\Sigma_\lambda^{-1} - \Sigma_\lambda^{-1}\Sigma\Sigma_\lambda^{-1})V^T$. $\hat{h}_i\hat{h}_j = (1/4)(\hat{h}_i+\hat{h}_j)^2 - (1/4)(\hat{h}_i-\hat{h}_j)^2$ with

$$\begin{aligned} \hat{h}_i + \hat{h}_j &= \frac{1}{n}\sum_{l=1}^{n}\left(\varphi(x_{p,l}, c_i) + \varphi(x_{p,l}, c_j)\right) \\ &\quad - \frac{1}{m}\sum_{l=1}^{m}\left(\varphi(x_{q,l}, c_i) + \varphi(x_{q,l}, c_j)\right) \\ \hat{h}_i - \hat{h}_j &= \frac{1}{n}\sum_{l=1}^{n}\left(\varphi(x_{p,l}, c_i) - \varphi(x_{q,l}, c_j)\right) \\ &\quad - \frac{1}{m}\sum_{l=1}^{m}\left(\varphi(x_{p,l}, c_i) - \varphi(x_{q,l}, c_j)\right) \end{aligned}$$

where $\varphi(x_l, c_i) = \exp(-[(||x_l - c_i||_2^2)/2\sigma^2])$. Assume $\varphi(x_{p,l}, c_i) + \varphi(x_{p,l}, c_j)$, $\varphi(x_{q,l}, c_i) + \varphi(x_{q,l}, c_j)$, $\varphi(x_{p,l}, c_i) -$

$\varphi(x_{q,l}, c_j)$, and $\varphi(x_{p,l}, c_i) - \varphi(x_{q,l}, c_j)$ follow distributions with the means and variances as $(u_{pij+}, \delta^2_{pij+})$, $(u_{qij+}, \delta^2_{qij+})$, $(u_{pij-}, \delta^2_{pij-})$, and $(u_{qij-}, \delta^2_{qij-})$, respectively. The central limit theorem guarantees that, when n and m are sufficiently large, the following terms converge to Gaussian distributions:

$$\frac{1}{n}\sum_{l=1}^{n}\left(\varphi\left(x_{p,l}, c_i\right) + \varphi\left(x_{p,l}, c_j\right)\right) \xrightarrow{d} N\left(\mu_{pij+}, \frac{\delta^2_{pij+}}{n}\right)$$
$$\frac{1}{m}\sum_{l=1}^{m}\left(\varphi\left(x_{q,l}, c_i\right) + \varphi\left(x_{q,l}, c_j\right)\right) \xrightarrow{d} N\left(\mu_{qij+}, \frac{\delta^2_{qij+}}{m}\right)$$
$$\frac{1}{n}\sum_{l=1}^{n}\left(\varphi\left(x_{p,l}, c_i\right) - \varphi\left(x_{q,l}, c_j\right)\right) \xrightarrow{d} N\left(\mu_{pij-}, \frac{\delta^2_{pij-}}{n}\right)$$
$$\frac{1}{m}\sum_{l=1}^{m}\left(\varphi\left(y_{p,l}, c_i\right) - \varphi\left(y_{q,l}, c_j\right)\right) \xrightarrow{d} N\left(\mu_{qij-}, \frac{\delta^2_{qij-}}{m}\right).$$

Considering the case with fixed kernel centers $\{c_i, i = 1, \ldots, k\}$, matrix H as well as H_λ^{-1} are fixed. In this case, since variables x_p and x_q are independent and the exponential functions are measurable, $\hat{h}_i + \hat{h}_j$ and $\hat{h}_i - \hat{h}_j$ also follow gaussian distributions:

$$\hat{h}_i + \hat{h}_j \xrightarrow{d} N\left(\mu_{pij+} - \mu_{qij+}, \frac{\delta^2_{pij+}}{n} + \frac{\delta^2_{qij+}}{m}\right)$$
$$\hat{h}_i - \hat{h}_j \xrightarrow{d} N\left(\mu_{pij-} - \mu_{qij-}, \frac{\delta^2_{pij-}}{n} + \frac{\delta^2_{qij-}}{m}\right).$$

Therefore, $(\hat{h}_i + \hat{h}_j)^2/([(\delta^2_{pij+})/n] + [(\delta^2_{qij+})/m])$ and $(\hat{h}_i - \hat{h}_j)^2/([(\delta^2_{pij+})/n] + [(\delta^2_{qij+})/m])$ are noncentral Chi-square distributed with 1 degree of freedom. Since we have established the preliminary results, the proof of Theorem 1 is as follows.

Proof (Theorem 1): $\hat{h}_i\hat{h}_j = (1/4)(\hat{h}_i+\hat{h}_j)^2-(1/4)(\hat{h}_i-\hat{h}_j)^2$ is distributed as a combination of two nonindependent noncentral Chi-square distributions.

Following the symmetry of matrix H, we have $(H_\lambda^{-1})^T = 2(H+\lambda I)^{-T} - (H+\lambda I)^{-1}H^T(H+\lambda I)^{-T} = H_\lambda^{-1}$ so that H_λ^{-1} is symmetric.

As a consequence, $\hat{D}^2_\lambda(p,q) = \sum_{i=1}^{k}\sum_{j=1}^{k} H^{-1}_{\lambda(i,j)}\hat{h}_i\hat{h}_j$ is distributed as a linear combination of $k(k+1)$ nonindependent noncentral Chi-square distributions.

APPENDIX C
PROOF OF THEOREM 2

Expectation With Finite Samples

Based on the analysis in Appendix B, we compute the expectations

$$E\left(\left(\hat{h}_i + \hat{h}_j\right)^2\right) = D\left(\hat{h}_i + \hat{h}_j\right) + E^2\left(\hat{h}_i + \hat{h}_j\right) = \frac{\delta^2_{pij+}}{n} + \frac{\delta^2_{qij+}}{m} + \left(\mu_{pij+} - \mu_{qij+}\right)^2$$
$$E\left(\left(\hat{h}_i - \hat{h}_j\right)^2\right) = D\left(\hat{h}_i + \hat{h}_j\right) + E^2\left(\hat{h}_i + \hat{h}_j\right) = \frac{\delta^2_{pij-}}{n} + \frac{\delta^2_{qij-}}{m} + \left(\mu_{pij-} - \mu_{qij-}\right)^2$$

$$E\left(\hat{h}_i\hat{h}_j\right) = \frac{1}{4}E\left(\left(\hat{h}_i + \hat{h}_j\right)^2\right) - \frac{1}{4}E\left(\left(\hat{h}_i - \hat{h}_j\right)^2\right) = \frac{1}{4}\left(\left(\frac{\delta^2_{pij+}}{n} + \frac{\delta^2_{qij+}}{m}\right) + \left(\mu_{pij+} - \mu_{qij+}\right)^2 - \left(\frac{\delta^2_{pij-}}{n} + \frac{\delta^2_{qij-}}{m}\right) - \left(\mu_{pij-} - \mu_{qij-}\right)^2\right).$$

Proof [Theorem 2 1)]: When x_p and x_q are generated from the same distribution, i.e., under H_0 with $p(x) = q(x)$, we have $\delta^2_{pij+} = \delta^2_{qij+}$, $\delta^2_{pij-} = \delta^2_{qij-}$, $\mu_{pij+} = \mu_{qij+}$, and $\mu_{pij-} = \mu_{qij-}$. Therefore

$$E_{H_0}\left(\hat{D}^2_\lambda(p,q)\right) = \sum_{i=1}^{k}\sum_{j=1}^{k} H^{-1}_{\lambda(i,j)}E\left(\hat{h}_i\hat{h}_j\right) = \frac{1}{4}\left(\frac{1}{n}+\frac{1}{m}\right)\sum_{i=1}^{k}\sum_{j=1}^{k} H^{-1}_{\lambda(i,j)}\left(\delta^2_{pij+} - \delta^2_{pij-}\right)$$

which shows that by increasing sizes n and m, the expectation values of estimated LSDD values under H_0 decrease. When H_1 holds, i.e., $p(x) \neq q(x)$

$$E_{H_1}\left(\hat{D}^2_\lambda(p,q)\right) = \frac{1}{4}\sum_{i=1}^{k}\sum_{j=1}^{k} H^{-1}_{\lambda(i,j)}\left(\left(\frac{\delta^2_{pij+}}{n} + \frac{\delta^2_{qij+}}{m}\right) + \left(\mu_{pij+} - \mu_{qij+}\right)^2 - \left(\frac{\delta^2_{pij-}}{n} + \frac{\delta^2_{qij-}}{m}\right) - \left(\mu_{pij-} - \mu_{qij-}\right)^2\right)$$

which indicates in the nonstationary conditions, the expectation values of $\hat{D}^2_\lambda$ also decrease with the increase of n and m.

Proof [Theorem 2 2)]: The difference

$$E_{H_1}\left(\hat{D}^2_\lambda(p,q)\right) - E_{H_0}\left(\hat{D}^2_\lambda(p,q)\right) = \frac{1}{4}\sum_{i=1}^{k}\sum_{j=1}^{k} H^{-1}_{\lambda(i,j)}\left(\frac{1}{m}\left(\delta^2_{qij+} - \delta^2_{pij+} - \delta^2_{qij-} + \delta^2_{pxj-}\right) + \left(\mu_{pij+} - \mu_{qij+}\right)^2 - \left(\mu_{pij-} - \mu_{qij-}\right)^2\right) \tag{27}$$

which is independent of n, but on m only. In other words, $E_{H_1}(\hat{D}^2_\lambda(p,q))$ decreases with the increase of n, whereas the difference between the expectations under H_0 and H_1 is fixed no matter how n varies.

Deviation Bounds for Tests With Finite Samples

In order to obtain the upper bound of the difference between $\hat{D}^2_\lambda(p,q)$ and its expectation $E(\hat{D}^2_\lambda(p,q))$ under both hypotheses H_0 and H_1, we apply the McDiarmid bound [38], [39] on the estimate $\hat{D}^2_\lambda(p,q)$.

Proof [Theorem 2 3)]: Let $n+m$ independent variables $x_{p,1}, \ldots, x_{p,n}, x_{q,1}, \ldots, x_{q,m}$ sampled from some set A, and assume that $\hat{D}^2_\lambda(p,q) = f(x_{p,1}, \ldots, x_{p,n}, x_{q,1}, \ldots, x_{q,m})$: $A^{m+n} \to R$. Changing either $x_{p,l}$ or $x_{q,l}$ in f results in changes

of $\hat{h}$ of at most $(1/n)$ or $(1/m)$. We first consider the case with $\hat{h}'_i = \hat{h}_i + (1/n)$, and $-1 \le \hat{h}_i, \hat{h}'_i \le 1$

$$\begin{aligned}
&\sup_{x_{p,1},\ldots,x_{q,m},\hat{x}_{p,l}} \left|\hat{D}^2_\lambda(p,q) - \hat{D}'^2_\lambda(p,q)\right| \\
&\quad = \left|\hat{h}^T H_\lambda^{-1}\hat{h} - \hat{h}'^T H_\lambda^{-1}\hat{h}'\right| \\
&\quad = \left|\sum_{i=1}^{k}\sum_{j=1}^{k} H^{-1}_{\lambda(i,j)}\hat{h}_i\hat{h}_j \right. \\
&\qquad \left. - \sum_{i=1}^{k}\sum_{j=1}^{k} H^{-1}_{\lambda(i,j)}\left(\hat{h}_i + \frac{1}{n}\right)\left(\hat{h}_j + \frac{1}{n}\right)\right| \\
&\quad = \left|\sum_{i=1}^{k}\sum_{j=1}^{k}\left(\frac{2}{n}\hat{h}_i H^{-1}_{\lambda(i,j)} + \frac{1}{n^2}H^{-1}_{\lambda(i,j)}\right)\right| \\
&\quad \le \frac{2}{n}\sum_{i=1}^{k}\sum_{j=1}^{k}\left|\hat{h}_i\right|\left|H^{-1}_{\lambda(i,j)}\right| + \frac{1}{n^2}\left|\sum_{i=1}^{k}\sum_{j=1}^{k} H^{-1}_{\lambda(i,j)}\right|.
\end{aligned}$$

H_λ^{-1} can be expressed as

$$H_\lambda^{-1} = V\begin{pmatrix} \frac{1}{\hat{r}_1} & & & \\ & \frac{1}{\hat{r}_2} & & \\ & & \cdots & \\ & & & \frac{1}{\hat{r}_k} \end{pmatrix} V^T \tag{28}$$

and $H^{-1}_{\lambda(i,j)} = \sum_{l=1}^{k}(1/\hat{r}_l)V_{i,l}V_{j,l}$. Since V is an orthogonal matrix, it is obvious that $(1/\hat{r}_1) < H^{-1}_{\lambda(i,i)} < (1/\hat{r}_k)$, $i = 1,\ldots,k$. We then estimate the bound of $H^{-1}_{\lambda(i,j)}$ as

$$\begin{aligned}
H^{-1}_{\lambda(i,i)} + H^{-1}_{\lambda(j,j)} + 2H^{-1}_{\lambda(i,j)} &= \sum_{l=1}^{k}\frac{1}{\hat{r}_l}\left(V_{i,l} + V_{j,l}\right)^2 \\
&< \frac{1}{\hat{r}_k}\sum_{l=1}^{k}\left(V_{i,l} + V_{j,l}\right)^2 = \frac{2}{\hat{r}_k} \\
H^{-1}_{\lambda(i,i)} + H^{-1}_{\lambda(j,j)} + 2H^{-1}_{\lambda(i,j)} &> \frac{1}{\hat{r}_1}\sum_{l=1}^{k}\left(V_{i,l} + V_{j,l}\right)^2 = \frac{2}{\hat{r}_1}.
\end{aligned}$$

Therefore, $(2/\hat{r}_1) < H^{-1}_{\lambda(i,i)} + H^{-1}_{\lambda(j,j)} + 2H^{-1}_{\lambda(i,j)} < (2/\hat{r}_k)$, and we can derive that $|H^{-1}_{\lambda(i,j)}| < (1/\hat{r}_k) - (1/\hat{r}_1) < (1/\hat{r}_k) < (2/\lambda)$. In addition

$$\begin{aligned}
\left|\sum_{i=1}^{k}\sum_{j=1}^{k} H^{-1}_{\lambda(i,j)}\right| &= \left|\sum_{i=1}^{k}\frac{1}{\hat{r}_i}\left(\sum_{j=1}^{k} V_{ji}\right)^2\right| \\
&< \left|\frac{1}{\hat{r}_k}\sum_{i=1}^{k}\left(\sum_{j=1}^{k} V_{ji}\right)^2\right| \\
&= \frac{k}{\hat{r}_k} < \frac{2k}{\lambda}.
\end{aligned}$$

The upper extreme satisfies

$$\begin{aligned}
&\sup_{x_{p,1},\ldots,x_{q,m},\hat{x}_{p,l}} \left|\hat{D}^2_\lambda(p,q) - \hat{D}'^2_\lambda(p,q)\right| \\
&\qquad < \frac{2k^2}{n}\frac{2}{\lambda} + \frac{1}{n^2}\frac{2k}{\lambda} = \frac{2k(2nk+1)}{n^2\lambda}.
\end{aligned}$$

The same procedure can be applied when considering $\hat{h}'_i = \hat{h}_i + (1/m)$, and we obtain

$$\sup_{x_{p,1},\ldots,x_{q,m},\hat{x}_{q,l}} |\hat{D}^2_\lambda(p,q) - \hat{D}'^2_\lambda(p,q)| < \frac{2k(2mk+1)}{m^2\lambda}.$$

Consequently, according to McDiarmid's Inequality [38], [39], for any $\epsilon > 0$, we have

$$\begin{aligned}
&\Pr\left(\left|\hat{D}^2_\lambda(p,q) - E\left(\hat{D}^2_\lambda(p,q)\right)\right| \ge \epsilon\right) \\
&\qquad \le \exp\left(-\frac{\epsilon^2\lambda^2}{\frac{4k^2(2nk+1)^2}{n^3} + \frac{4k^2(2mk+1)^2}{m^3}}\right)
\end{aligned}$$

where Pr denotes the probability over n samples with pdf $p(x)$ and m with $q(x)$.

The inequality reveals some properties:

1) with the increase of sizes n and m, the deviation bounds decrease;
2) a larger λ or a smaller k (less centers) will bring a larger deviation bound;
3) without adding other restrictions, the above inequality applies to both cases under H_0 and H_1.

REFERENCES

[1] H. Liu, F. Sun, D. Guo, B. Fang, and Z. Peng, "Structured output-associated dictionary learning for haptic understanding," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2016.2635141.

[2] H. Liu, J. Qin, F. Sun, and D. Guo, "Extreme kernel sparse learning for tactile object recognition," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2016.2614809.

[3] H. Liu, D. Guo, and F. Sun, "Object recognition using tactile measurements: Kernel sparse coding methods," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 656–665, Mar. 2016.

[4] H. Liu, Y. Liu, and F. Sun, "Robust exemplar extraction using structured sparse coding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1816–1821, Aug. 2015.

[5] Q. Zhang, D. Zhao, and Y. Zhu, "Event-triggered H_∞ control for continuous-time nonlinear system via concurrent learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2016.2531680.

[6] Q. Zhang, D. Zhao, and D. Wang, "Event-based robust control for uncertain nonlinear systems using adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2016.2614002.

[7] C. Alippi and M. Roveri, "Just-in-time adaptive classifiers—Part I: Detecting nonstationary changes," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1145–1153, Jul. 2008.

[8] C. Alippi, G. Boracchi, and M. Roveri, "Just in time classifiers: Managing the slow drift case," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Atlanta, GA, USA, 2009, pp. 114–120.

[9] D. M. Hawkins and K. Zamba, "Statistical process control for shifts in mean or variance using a changepoint formulation," *Technometrics*, vol. 47, no. 2, pp. 164–173, 2005.

[10] G. J. Ross, D. K. Tasoulis, and N. M. Adams, "Nonparametric monitoring of data streams for changes in location and scale," *Technometrics*, vol. 53, no. 4, pp. 379–389, 2011.

[11] G. J. Ross and N. M. Adams, "Two nonparametric control charts for detecting arbitrary distribution changes," *J. Qual. Technol.*, vol. 44, no. 2, pp. 102–116, 2012.

[12] D. M. Hawkins, P. Qiu, and C. W. Kang, "The changepoint model for statistical process control," *J. Qual. Technol.*, vol. 35, no. 4, pp. 355–366, 2003.

[13] Y. Lepage, "Combination of Wilcoxians and Ansari–Bradley statistics," *Biometrika*, vol. 58, no. 1, pp. 213–217, Apr. 1971.
[14] M. F. Schilling, "Multivariate two-sample tests based on nearest neighbors," *J. Amer. Stat. Assoc.*, vol. 81, no. 395, pp. 799–806, 1986.
[15] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Online outlier detection in sensor data using non-parametric models," in *Proc. Int. Conf. Very Large Data Bases (VLDB)*, Seoul, South Korea, 2006, pp. 187–198.
[16] R. Sebastião, J. Gama, P. P. Rodrigues, and J. Bernardes, "Monitoring incremental histogram distribution for change detection in data streams," in *Proc. 2nd Int. Conf. Knowl. Discov. Sensor Data*, Las Vegas, NV, USA, 2010, pp. 25–42.
[17] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi, "An information-theoretic approach to detecting changes in multi-dimensional data streams," in *Proc. Symp. Interface Stat. Comput. Sci. Appl.*, vol. 67. Pasadena, CA, USA, 2006, pp. 1–21.
[18] Q. Xu and L. J. Karam, "Change detection on SAR images by a parametric estimation of the KL-divergence between Gaussian mixture models," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, Canada, 2013, pp. 2109–2113.
[19] G. Ditzler and R. Polikar, "Hellinger distance based drift detection for nonstationary environments," in *Proc. IEEE Symp. Comput. Intell. Dyn. Uncertain Environ.*, Paris, France, 2011, pp. 41–48.
[20] C. Alippi, G. Boracchi, D. Carrera, and M. Roveri, "Change detection in multivariate datastreams: Likelihood and detectability loss," unpublished paper, 2015. [Online]. Available: https://arxiv.org/abs/1510.04850
[21] 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, Jul. 2013.
[22] M. Sugiyama *et al.*, "Density-difference estimation," *Neural Comput.*, vol. 25, no. 10, pp. 2734–2775, 2013.
[23] L. Bu, C. Alippi, and D. Zhao, "A pdf-free change detection test based on density difference estimation," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2016.2619909.
[24] L. Bu, C. Alippi, and D. Zhao, "Ensemble LSDD-based change detection tests," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Vancouver, BC, Canada, 2016, pp. 4064–4069.
[25] C. Alippi, G. Boracchi, and M. Roveri, "A just-in-time adaptive classification system based on the intersection of confidence intervals rule," *Neural Netw.*, vol. 24, no. 8, pp. 791–800, 2011.
[26] A. C. Davison and D. V. Hinkley, *Bootstrap Methods and Their Application*, vol. 1. Cambridge, U.K.: Cambridge Univ. Press, 1997.
[27] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, pp. 37–57, 1985.
[28] H. Raza, G. Prasad, and Y. Li, "EWMA model based shift-detection methods for detecting covariate shifts in non-stationary environments," *Pattern Recognit.*, vol. 48, no. 3, pp. 659–669, 2015.
[29] G. Ditzler and R. Polikar, "Hellinger distance based drift detection for nonstationary environments," in *Proc. IEEE Symp. Comput. Intell. Dyn. Uncertain Environ.*, Paris, France, 2011, pp. 41–48.
[30] L. L. Minku, A. P. White, and X. Yao, "The impact of diversity on online ensemble learning in the presence of concept drift," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 5, pp. 730–742, May 2010.
[31] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Mach. Learn.*, vol. 23, no. 1, pp. 69–101, 1996.
[32] P. Tüfekci, "Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods," *Int. J. Elect. Power Energy Syst.*, vol. 60, pp. 126–140, Sep. 2014.
[33] H. Kaya, P. Tüfekci, and F. S. Gürgen, "Local and global learning methods for predicting power of a combined gas & steam turbine," in *Proc. Int. Conf. Emerg. Trends Comput. Electron. Eng.*, Dubai, UAE, 2012, pp. 13–18.
[34] C. Alippi, D. Liu, D. Zhao, and L. Bu, "Detecting and reacting to changes in sensing units: The active classifier case," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 3, pp. 353–362, Mar. 2014.
[35] C. Alippi, G. Boracchi, and M. Roveri, "Change detection tests using the ICI rule," in *Proc. Int. Joint Conf. Neural Netw.*, Barcelona, Spain, 2010, pp. 1–7.
[36] A. Gretton *et al.*, "Optimal kernel choice for large-scale two-sample tests," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1205–1213.
[37] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
[38] C. McDiarmid, "On the method of bounded differences," *Surveys Comb.*, vol. 141, no. 1, pp. 148–188, 1989.
[39] K. Sridharan, "A gentle introduction to concentration inequalities," Dept. Comput. Sci., Cornell Univ., Tech. Rep., 2002. [Online]. Available: http://www.cs.cornell.edu/∼sridharan/concentration.pdf

Li Bu received the B.S. degree in electronic engineering and automation from the China University of Mining and Technology, Xuzhou, China, in 2012. She is currently pursuing the Ph.D. degree in computer science with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences and University of Chinese Academy of Sciences, Beijing, China.

Her current research interests include adaptation and learning in nonstationary environments and computational intelligence.

Dongbin Zhao (M'06–SM'10) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1994, 1996, and 2000, respectively.

He was a Post-Doctoral Fellow with Tsinghua University, Beijing, China, from 2000 to 2002. He has been a Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing, since 2002, and also a Professor with the University of Chinese Academy of Sciences, Beijing. From 2007 to 2008, he was also a Visiting Scholar with the University of Arizona, Tucson, AZ, USA. He has published four books and over 60 international journal papers. His current research interests include computational intelligence, adaptive dynamic programming, deep reinforcement learning, robotics, intelligent transportation systems, and smart grids.

Dr. Zhao has been an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS since 2012, and the IEEE COMPUTATION INTELLIGENCE MAGAZINE since 2014. He is the Chair of Beijing Chapter, and was the Chair of Adaptive Dynamic Programming and Reinforcement Learning Technical Committee from 2015 to 2016, Multimedia Subcommittee from 2015 to 2016 of IEEE Computational Intelligence Society. He works as several guest editors of renowned international journals. He is involved in organizing several international conferences.

Cesare Alippi (SM'94–F'06) received the (*cum laude*) degree in electronic engineering and the Ph.D. degree from the Politecnico di Milano, Milan, Italy, in 1990 and 1995, respectively.

He is currently a Full Professor with the Politecnico di Milano, and Università della Svizzera italiana, Lugano, Switzerland. He has been a Visiting Researcher with University College London, London, U.K., the Massachusetts Institute of Technology, Cambridge, MA, USA, ESPCI, Paris, France, CASIA, Beijing, China, A*STAR, Singapore, and UKobe, University of Kobe, Kobe, Japan. He holds five patents, has published one monograph book, six edited books, and about 200 papers in international journals and conference proceedings. His current research interests include adaptation and learning in nonstationary environments and intelligence for embedded and cyber-physical systems.

Dr. Alippi was a recipient of the Gabor Award from the International Neural Networks Society and the IEEE Computational Intelligence Society Outstanding Transactions on Neural Networks and Learning Systems Paper Award in 2016, the IBM Faculty Award in 2013; the IEEE Instrumentation and Measurement Society Young Engineer Award in 2004. He is a Board of Governors Member of the International Neural Network Society and the European Neural Network Society, the Past Vice-President Education of the IEEE Computational Intelligence Society, the Past Associate Editor of the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE, the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENTS, the IEEE TRANSACTIONS ON NEURAL NETWORKS.