A Kolmogorov-Smirnov Test to Detect Changes in Stationarity in Big Data ⋆

Dongbin Zhao * **Li Bu** * **Cesare Alippi** ** **Qinglai Wei** *

* *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: dongbin.zhao@ia.ac.cn, bulipolly@gmail.com, qinglai.wei@ia.ac.cn).*
** *The Politecnico di Milano, Italy and Università della Svizzera italiana, Switzerland, (e-mail: cesare.alippi@polimi.it)*

Abstract: The paper proposes an effective change detection test for online monitoring data streams by inspecting the least squares density difference (LSDD) features extracted from two non-overlapped windows. The first window contains samples associated with the pre-change probability distribution function (pdf) and the second one with the post-change one (that differs from the former if a change in stationarity occurs). This method can detect changes by also controlling the false positive rate. However, since the window sizes is fixed after the test has been configured (it has to be small to reduce the execution time), the method may fail to detect changes with small magnitude which need more samples to reach the requested level of confidence. In this paper, we extend our work to the Big Data framework by applying the Kolmogorov-Smirnov test (KS test) to infer changes. Experiments show that the proposed method is effective in detecting changes.

Keywords: change detection test, LSDD, KS test.

1. INTRODUCTION

In the framework of Big Data, the data generating process may undergo concept drift (changes in stationarity) since it is unlikely that the pdf will not change in the long run, e.g., due to aging effects affecting the sensor, etc. This non-stationarity changes the underlying distribution of data, and results in performance degradation of the application processing sensor data. This follows from the fact that models/applications trained with pre-change data (that always assume the stationary hypothesis) are no more working on the operational conditions when changes occur. To deal with such issues, one effective solution is to detect concept drifts, and then retrain models/applications with data associated with the changed pdf. This is what we call “active approach”(Alippi et al. (2013)).

For online change detection, window-based methods are generally preferred to keep under control computational time (Dasu et al. (2006); Sebastião et al. (2010); Alippi et al. (2015); Chen et al. (2014)). In general, two non-overlapped windows are used to represent the pre-change and the possibly post-change conditions, respectively. The first window Z_p contains n_0 stationary samples and constitutes the reference set, generated according to pdf $p(x)$. The testing set Z_q slides to collect newly arrived n_1 samples generated by pdf $q(x)$. Change in stationarity occurs when $q(x) \neq p(x)$. From the operational point of view changes are detected when the statistics evaluated over the two windows exceed a threshold function of a confidence level.

In our previous work(Bu et al. (2015, 2016b)), we proposed an effective change detection test (CDT) based on the least squares density difference (LSDD). The test directly measures the density-difference of the pdfs(Sugiyama et al. (2013)) with a linear-in-parameters Gaussian kernel function, which reduces the estimation error associated with estimating the two pdfs individually. Obtained LSDD features get larger as $p(x)$ drifts from $q(x)$. We detect changes by comparing the LSDD values with a threshold associated with a predefined confidence level of detection, i.e., the required false positive (FP) rate (Bu et al. (2016b)).

In general, LSDD-based methods operate under the condition that only a limited data set is provided for training the detection method. In the extreme case, a bootstrap procedure is applied to assess the empirical distribution of LSDD values (Bu et al. (2016b)). In order to make it more sensitive to changes, the reservoir sampling strategy for updating the reference set Z_p and a three-level mechanism are adopted to reduce the false negative (FN) rates (Bu et al. (2016b)). Such effort also includes exploring different detecting strategies with ensemble learning (Bu et al. (2016a)). Nevertheless, they still show poor performance with too high FN rates when handling changes with small magnitude (Bu et al. (2016b)).

When considering the Big Data framework, the limitation of data disappears and there are enough training data

⋆ This work was supported in part by the National Natural Science Foundation of China under Grants No. 61573353, No.61533017, and No. 61603382.

to be used to configure change detection based on large windows of data. As such, it is possible to detect smaller changes. However, larger window sizes correspond to a higher computational time which might make the method unable to satisfy the online real-time constraints for mid-size computational devices, e.g., those used in the Internet-of-Things. Furthermore, the sizes can not adapt according to the magnitude of changes, since thresholds are derived under the constraint of fixed window sizes.

In this paper, we propose a prior-free change detection test by combining the LSDD CDT with the Kolmogorov-Smirnov test (KS test). The test windows corresponding to the LSDD method are independent and non-overlapped, which also grants independent and one-dimensional LS-DD statistics. We then measure the dissimilarity of the distributions of estimated LSDD values with a KS test, instead of comparing them with a threshold as usual. Since the execution of KS test is simple and their statistics are distributed according to a known Kolmogorov distribution (Wang et al. (2003)), it can be applied here to infer changes with more samples and keep the LSDD-associated window sizes small.

2. THE DISTRIBUTION OF LSDD VALUES

2.1 The LSDD Method

The least squares density difference estimation between two pdfs was firstly proposed in (Sugiyama et al. (2013)):

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

where $x \in R^d$ is a real vector, and $p(x), q(x)$ represent two pdfs generating the reference set Z_p and testing set Z_q respectively. Kernel function

$$g(x,\Theta) = \sum_{i=1}^{n_c} \theta_i \exp(-\frac{||x-c_i||_2^2}{2\sigma^2}),\ \Theta{=}(\theta_1,...\theta_i,...\theta_{n_c}) \tag{2}$$

is used to estimate $p(x) - q(x)$, where $\{c_i, i = 1, ..., n_c\}$ are d-dimensional kernel centers, randomly chosen from the training set. Θ is a parameter vector, and $\sigma = median(||x_i -_j ||_2, 0 < i < j \leq N_t)$ is the scale parameter (Gretton et al. (2012)) where N_t is the cardinality of the training set. The optimal parameter Θ^* is achieved by minimizing the loss $J(\Theta)$ with regularization:

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

where $\lambda > 0$. After some derivations, please refer to (Bu et al. (2015, 2016b)) for details, we have:

$$\hat{D}^2(p,q) = 2\hat{h}^T\hat{\Theta} - \hat{\Theta}^T H\hat{\Theta}, \tag{4}$$

where $\hat{D}^2$ is the estimate of D^2, $\hat{\Theta} = (H + \lambda I)^{-1}\hat{h}$, H is an $n_c \times n_c$ matrix, and $\hat{h}$ is an $n_c \times 1$ vector:

$$H_{i,j} = (\pi\sigma^2)^{d/2} \exp(-\frac{||c_i - c_j||_2^2}{4\sigma^2}), \tag{5}$$

$$\begin{aligned} \hat{h}_i =& \frac{1}{n_0}\sum_{j=1}^{n_0} \exp(-\frac{||x_{p,j} - c_i||_2^2}{2\sigma^2}) \\ &- \frac{1}{n_1}\sum_{j=1}^{n_1} \exp(-\frac{||x_{q,j} - c_i||_2^2}{2\sigma^2}), \end{aligned} \tag{6}$$

$i = 1, ..., n_c$. x_p, x_q represent instances from sets Z_p and Z_q so that $Z_p = \{x_{p,j}, j = 1, ..., n_0\}$, $Z_q = \{x_{q,j}, j = 1, ..., n_1\}$.

It's worth mentioning that parameter λ controls the s-moothness of approximating model $g(x, \hat{\Theta})$ and, as such, influences the detection performance of LSDD CDT. Here we choose λ as suggesting in (Bu et al. (2015, 2016b)).

2.2 The Distribution of LSDD Values

We have proved in (Bu et al. (2016c)) that when the kernel centers in (2) are fixed, the distribution Π_0 of estimated LSDD values in stationarity conditions is a linear combination of $n_c(n_c + 1)$ non-central chi-square distributions. Since the distribution requires too many parameters to be estimated, we suggest to derive thresholds directly as the $1-\mu$ percentile of the estimates (Bu et al. (2016c)), where μ represents the user-defined FP rate.

However, the distribution is controlled by window sizes n_0 and n_1. That is, once we enlarge or shrink a window, re-training is needed to rebuild a new empirical distribution.

3. LSDD-KS

The detection strategy of LSDD transforms multidimensional input data characterized by an unknown pdf into one-dimensional feature: by verifying if the new LSDD value comes from distribution Π_0, changes may be detected. Somehow, it is like recognizing outliers. As we mentioned, the effectiveness of the test depends on the size of the data window. If the window size is small, we will end up with high false positives (FPs) and false negatives (FNs).

In order to mitigate this problem, two solutions can be considered: one is to use larger windows; the other suggests to test whether the distribution of possible outliers is the same as Π_0 or not. For the former case, a fixed large window means more cost in time, which may be inoperable in the online monitoring of Big Data. For the latter one, it's actually a one-dimensional change detection problem, where many mature and simple methods can be easily applied, such as the change point models (CPMs) (Ross et al. (2011); Ross and Adams (2012)) and the KS test (Massey Jr (1951)).

The two-sample KS test could be one of the most general nonparametric tests. It can detect the differences in both location and shape of the empirical cumulative distribution functions of the two sample sets (two windows). Since the tables of critical values associated with two-sample KS test have been published, it can be applied easily without retraining. That is, the sizes of KS test can be variable.

In this paper, we propose the LSDD-KS CDT by combining the LSDD with two sample KS test working with two-level windows. The new method can easily utilize more samples when needed to be more sensitive to small changes. Still, no assumptions about the distributions or other priors are required. The only basis of change detection is that samples in the training set are in stationarity.

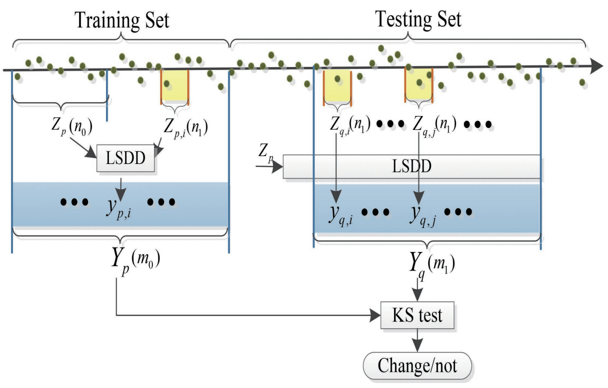

Fig. 1. The combined detection method of LSDD and KS test

3.1 Discussions about the Reference Set

As we have pointed out in (Bu et al. (2016c)), the reference set Z_p is a realization of a random variable and specific values influence the detection performance. In order to deal with randomness, reservoir sampling was suggested to update Z_p as done in (Bu et al. (2016b)), multi-sets were considered with ensemble learning in (Bu et al. (2016a)), and an enlarged window was applied during the testing phase only in (Bu et al. (2016c)).

Fortunately, when Big Data framework is available, a large reference set Z_p can be used. In particular, we randomly select n_0 samples (n_0 should be a large value) from the training set, and fix them during the whole training and testing phases until a change is detected.

3.2 The Algorithm of LSDD-KS

The detailed combining implementation of LSDD and the KS test is shown in Fig.1, where data window sizes n_i and $m_i\{i = 0, 1\}$ are associated with LSDD and KS test respectively. Here, the LSDD is employed to measure the density differences to derive the one-dimensional statistics D^2 (to ease understanding, we rename it as y_i); the change detection is then carried out by the KS test. In Fig.1, the yellow independent windows, each of which contains n_1 samples, are used to derive independent LSDD values, so that independent values $\{y_{p,i}, i = 1, ..., m_0\}$ and $\{y_{q,j}, j = 1, ..., m_1\}$ for the KS test are derived. The blue windows work at the KS-test level; the left window is fixed after training while the second one slides to collect new data.

The description of the whole detection process is given in Algorithm 1. During the training phase (steps 1-3), we prepare windows for LSDD (steps 1-2) and generate the reference set Y_p for KS test (step 3). During the testing phase (steps 6-16), once n_1 new samples are arrived, the *LSDD* operator (step 6) is applied on sets Z_p and $Z_{q,j}$ to derive a statistic $y_{q,j}$. Then, changes are detected according to the KS test, with the *KStest* operator working on sets Y_p and Y_q under the constraint of a fixed expected FP rate μ (steps 9-13). The output of *KStest* η (step 9) says that whether the null hypothesis that $p(x) = q(x)$ is rejected, i.e., changes are detected (step 10) or not.

Algorithm 1 LSDD-KS

Input: Training set Ω with size N_t, window sizes n_0, n_1, m_0, m_1, FP rate μ, kernel centers $c_i\{i = 1, ..., n_c\}$;
Output: Change location or no changes.

```
1:  Randomly select n_0 samples from Ω as the reference set Z_p;
2:  Build m_0 sets Z_{p,i}{i = 1,...,m_0}, each of which contains n_1 samples from Ω, to be compared with Z_p.
3:  Construct set Y_p that y_{p,i} = LSDD(Z_p, Z_{p,i}), {i = 1,...,m_0}, and parameters σ and λ are chosen as suggested in Section II;
4:  j = 1;
5:  while (1) do
6:      Collect n_1 newly arrived samples into set Z_q, j with no overlap, y_{q,j} = LSDD(Z_p, Z_{q,j});
7:      if j ≥ m_1 then
8:          Construct set Y_q = {y_{q,i}, i = j + 1 − m_1, ..., j};
9:          Execute the KS test η = KStest(Y_p, Y_q, μ);
10:         if η = 1 then
11:             A change is detected at the j × n_1-th sample;
12:             Break;
13:         end if
14:     else
15:         j = j + 1;
16:     end if
17: end while
```

It's worth noting that since the kernel centers are fixed before training, matrix H, as well as $(H + \lambda I)^{-1}$, can be computed in advance (and only once). The operations about the reference set Z_p can also be executed beforehand that

$$f_p = \frac{1}{n_0} \sum_{j=1}^{n_0} \exp(-\frac{||x_{p,j} - c_i||_2^2}{2\sigma^2}). \tag{7}$$

In this case, the remaining task of *LSDD* operator is only to deal with the information about $Z_{q,j}$ and then estimate the LSDD value according to (4).

4. EXPERIMENTS

In order to verify the effectiveness of proposed LSDD-KS test, we designed several experiments to reveal the influence of the window size as well as show how the method is sensitive to small changes.

The basic LSDD-CDT method without the higher-level KS test is considered as a reference CDT for comparison. For such a test, we use the same fixed Z_p set as in LSDD-KS, and slide the testing set Z_q to collect newly arrived samples at each time step. During the training phase, a bootstrap procedure is used as recommended in (Bu et al. (2016b)) to build the empirical distribution of LSDD values so as to derive a threshold T_μ satisfying

$$P_r(\hat{D}^2 > T_\mu) = \mu. \tag{8}$$

During the testing phase, the LSDD value is computed on sets Z_q and Z_p at each time step to obtain y which is then compared to T_μ to detect possible changes.

We consider four simulated applications. For comprehensive evaluation, both one-dimensional and multi-dimensional cases and different change types are involved.

- Application D1 refers to two cases with different changes affecting the stationary pdf $N(0, 0.5)$. Changes occur abruptly (1#) or drifting slowly (2#) with pdf shifting to $N(0.1, 0.5)$. Both of the cases represent scenarios of "small" changes because of their small magnitude of changes.
- Application D2 is a ten dimensional problem in (Raza et al. (2015)), with instances satisfying a multivariate normal distribution. The means are fixed at $u_{1,i} = u_{2,i} = 0$, and the covariance shifts from $\sigma_{1,ij(i=j)} = 0.5, \sigma_{1,ij(i\neq j)} = 0$ to $\sigma_{2,ij(i=j)} = 0.5, \sigma_{2,ij(i\neq j)} = 0.4$ abruptly where $i, j = 1, ..., 10$.
- Application D3 is a two-class rotating mixture of Gaussions (Ditzler and Polikar (2011)) with class centers shifting from $\mu_1 = [1/\sqrt{2}, 1/\sqrt{2}]$, $\mu_2 = [-1/\sqrt{2}, -1/\sqrt{2}]$ to $\mu_1 = [1/\sqrt{2}, -1/\sqrt{2}]$, $\mu_2 = [-1/\sqrt{2}, 1/\sqrt{2}]$. The covariances are fixed during the whole detection $\Sigma_1 = \Sigma_2 = [0.5, 0; 0, 0.5]$.
- Application D4 is a moving hyperplane problem (Minku et al. (2010)) that $x_{k+1} \leq -a_0 + \sum_{i=1}^{k} a_i x_i$. Changes are added as in (Bu et al. (2016b)) by changing the value of a_0 from -1 to -3.2. The other parameters are $k = 2$ and $a_1 = a_2 = 0.1$. x_1, x_2 are random variables uniformly distributed from interval $[0, 1]$ and x_3 from $[0, 5]$.

The user-defined parameters are fixed as follows. The size of the training set N_t is 20000, the relatively large n_0 for LSDD is 2000, m_0 for KS test is 500, m_1 is 100, and n_c is 100. λ is configured as in (Bu et al. (2016c)), including the predefined relative difference $RD_0 = 0.25$ and their optional values generated by a Matlab function logspace(-2, 1, 20). The size of testing set is 40000 and changes occur from the 20001-th sample on. The impact of window size n_1 will be discussed in the next subsection.

Other parameters for the basic LSDD should be under the constraint of LSDD-KS that a 20000 data set can not allow a large size of Z_q, e.g., a window with size 50×100 compared to LSDD-KS. In this case, we make a compromise that the size n_1 here is set to 500 which is neither small nor too large. The number of bootstraps for building the empirical distribution is 1000.

Three indexes are used to evaluate the performance of the proposed LSDD-KS CDT:

- False positive rate (FP(%)): it counts the rate of false detections.
- False negative rate (FN(%)): it counts the rate of undetected changes.
- Delay (in samples): it measures the detection delay. A delay is recorded only when the change is detected. Both the mean and the standard deviation (in parenthesis) of the delay values are computed.

4.1 The Influence of Window Sizes

In this section, we discuss the influence of window sizes through experiments on applications D1-4. Six cases are considered exploring different values of n_1 and μ as shown in Table 1. Both the small windows ($n_1 = 50$) and slightly large ones ($n_1 = 200$) are included.

Table 1. Testing the real FP rates

μ	n_1	D1	D2	D3	D4
1%	50	0.6%	0.2%	0.8%	0.4%
	100	0.6%	0.8%	1%	0.4%
	200	1.8%	2.4%	0.8%	0.8%
5%	50	4%	4.8%	5.2%	5.8%
	100	5%	5%	3.8%	5%
	200	5%	6.8%	8%	7.4%

Firstly, 500 independent tests are carried out to measure the real FP and FN rates. For measuring the FP rates, the testing samples are with the same pdfs as the training samples, whereas for the FN ones, they are with the changed pdfs as introduced above.

As shown in Table 1, for all the applications D1-4, with the increase of n_1 ($n_1 = 50$ and 100), the FP rates show no obvious trend. Their real computed FP rates are mainly consistent with the predefined values. However, for $n_1 = 200$, since a larger window needs more samples to derive independent statistics to train a test, the real FP rates here are a bit higher.

In the experiments, the FN rates for all the applications with different parameters are 0, which means that the method will not miss any changes. We do not record them in a table any more.

In the next experiment, we conduct 100 independent tests over data streams with overlapped windows of LSDD-KS to reveal the relationship between the detection delay and n_1. Here, the delay (both the mean and the standard deviation are measured) shown in the top rows in Table 2, counts the samples starting from sample 20001 where the change is applied. Furthermore, we also record the relative delay (= delay/n_1) shown in the second rows which represent the testing times of the KS test.

It emerges that for small changes (D1), most of delays increase with the increase of window size n_1 as expected, whereas the relative delay decreases. It's because that more samples in Z_q will reveal the changes on LSDD values in time, so that the KS test, operated on these values, can detect changes quickly with fewer relative delay. However, the real delay is higher. For the drift case (2#), more samples are needed than in D1(1#) before concept drift becomes significant (and, hence, perceivable by the method). For other applications (D2-4) with stronger changes, the required instances, for showing their significant changes with the same FP rates, are almost the same. Furthermore, for all the applications, higher FP rates correspond to smaller delays as expected.

We conclude that the proposed LSDD-KS method can work well with controllable FP rates even with small size n_1, e.g., $n_1 = 50$. In addition, it is always preferred that the small n_1 is associated with small delay.

Table 2. The Detection Delay with Different Window Sizes

μ	n_1	D1(1#)	D1(2#)	D2	D3	D4
1%	50	2863.5(1330.4) 57.3	13403.2(5068.4) 268.1	745(213.7) 14.9	754(179.5) 15.1	747(146.7) 15
	100	3406(1514) 34.1	14533(4143.9) 145.3	1519(298.4) 15.2	1499(342.4) 15	1455(340.3) 14.6
	200	5470.7(2992.7) 27.4	14501.3(3894.9) 72.5	2986(667.4) 14.9	2942(701.7) 14.7	2984(756.2) 14.9
5%	50	2240(1909.1) 44.8	8726.3(5456.1) 174.5	559(195.8) 11.8	554.5(216.5) 11.1	529.5(213.7) 10.6
	100	3160(1955.6) 31.6	10347.4(5516) 103.5	1147(379.6) 11.5	1119(373) 11.2	1171(353.4) 11.7
	200	3528(2209) 17.6	12289.6(5883.7) 61.4	2046(920.2) 10.2	2172(717.5) 10.9	2144(900.3) 10.7

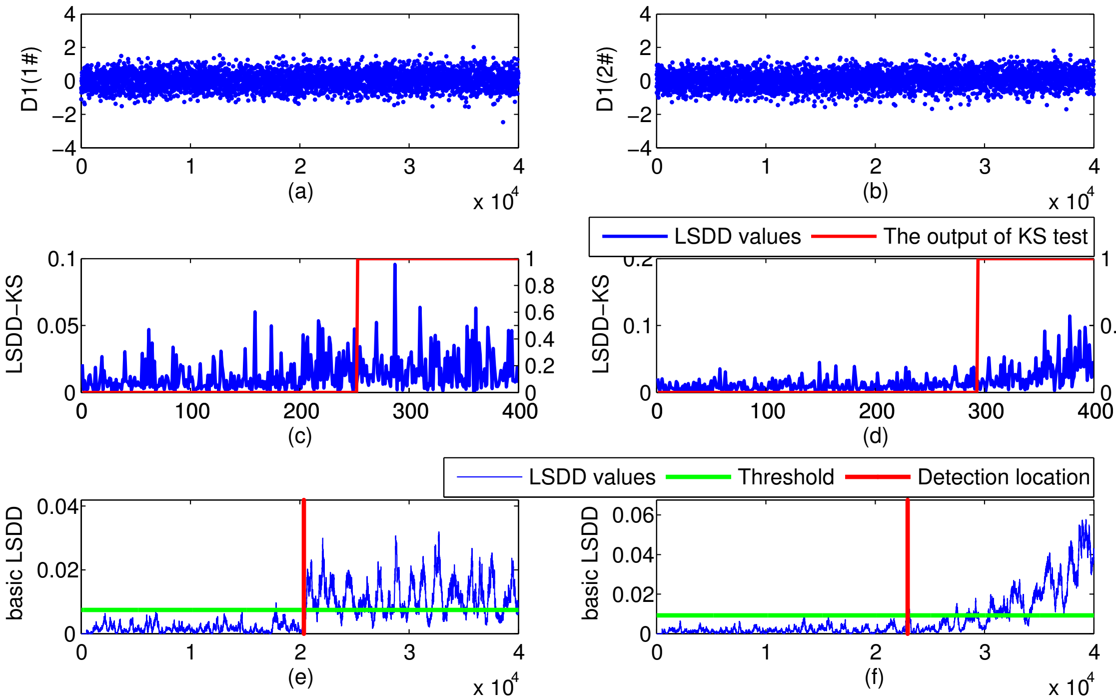

Fig. 2. The change detection performance of LSDD-KS and the basic LSDD on D1

4.2 The Change Detection Performance

In the last experiment, we make a comparison between LSDD-KS and the basic LSDD CDT on the two cases of application D1. The window size n_1 is set to 100 and the predefined FP rate μ is 1%. In order to observe how the derived LSDD statistics shift when changes occur, we run the detection test until the end of the data stream but not stop it as in Algorithm 1 (step 12).

Fig.2 (a) and (b) show the dataset of application D1; 1# is with an abrupt change and 2# with a slow drift. LSDD-KS detects both changes accurately (as shown in Fig.2 (c) and (d))with neither false positives nor false negatives at the cost of large delay. Even though the basic LSDD can detect changes more promptly, the false negatives in Fig.2 (e) and (f) are high (see the LSDD values under the threshold line after change location), which means it may always miss changes. Furthermore, there are also some false positives especially in Fig.2 (e) (see the LSDD values above the threshold line before change), which makes the method less credible.

5. CONCLUSIONS

In this paper, we propose a novel change detection test by combing the least squares density difference estimation with the KS test. In the framework of Big Data, independent windows are used to derive independent and one-dimensional LSDD values. Then the KS test is applied to detect possible changes which are reflected by the statistics. The method can work with small windows for LSDD which reduces the execution time.

The experiments validate that our method is effective in detecting changes with controllable FP rates. It has to be noted that, in this paper, we consider two cases with "small" changes which are hardly detected in our previous work even with severer changes in the magnitude. The results show that by having enough training data (Big Data framework), a small change can even be detected accurately.

REFERENCES

Alippi, C., Boracchi, G., Carrera, D., and Roveri, M. (2015). Change detection in multivariate datastreams: Likelihood and detectability loss. *arXiv preprint arXiv:1510.04850*.

Alippi, C., Bu, L., and Zhao, D. (2013). A prior-free encode-decode change detection test to inspect datastreams for concept drift. In *Proc. IEEE Int. Joint Conf. Neural Netw.*, 1–6.

Bu, L., Alippi, C., and Zhao, D. (2016a). Ensemble lsdd-based change detection tests. In *Proc. IEEE Int. Joint Conf. Neural Netw., to be published*. Vancouver, Canada.

Bu, L., Alippi, C., and Zhao, D. (2016b). A pdf-free change detection test based on density difference estimation. *IEEE Trans. Neural Netw. Learn. Syst.*

Bu, L., Zhao, D., and Alippi, C. (2015). A pdf-free change detection test for data streams monitoring. In *IEEE Symp. Ser. Comput. Intell.*, 542–547. doi: 10.1109/SSCI.2015.86.

Bu, L., Zhao, D., and Alippi, C. (2016c). An incremental change detection test based on density difference estimation. *IEEE Trans. Syst. Man Cybern. -Syst. under review*.

Chen, H., Tiňo, P., Rodan, A., and Yao, X. (2014). Learning in the model space for cognitive fault diagnosis. *IEEE Trans. Neural Netw. Learn. Syst.*, 25(1), 124–136.

Dasu, T., Krishnan, S., Venkatasubramanian, S., and Yi, K. (2006). An information-theoretic approach to detecting changes in multi-dimensional data streams. In *Proc. Symp. on the Interface of Statistics, Computing Science, and Applications*. Citeseer.

Ditzler, G. and Polikar, R. (2011). Hellinger distance based drift detection for nonstationary environments. In *IEEE Symp. on Comput. Intell. in Dynamic and Uncertain Environments*, 41–48. IEEE.

Gretton, A., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K., and Sriperumbudur, B.K. (2012). Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Process. Syst.*, 1205–1213.

Massey Jr, F.J. (1951). The kolmogorov-smirnov test for goodness of fit. *J. Am. Stat. Assoc.*, 46(253), 68–78.

Minku, L.L., White, A.P., and Yao, X. (2010). The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Trans. Knowl. Data Eng.*, 22(5), 730–742.

Raza, H., Prasad, G., and Li, Y. (2015). Ewma model based shift-detection methods for detecting covariate shifts in non-stationary environments. *Pattern Recognit.*, 48(3), 659–669.

Ross, G.J. and Adams, N.M. (2012). Two nonparametric control charts for detecting arbitrary distribution changes. *J. Qual. Technol.*, 44(2), 102–116.

Ross, G.J., Tasoulis, D.K., and Adams, N.M. (2011). Nonparametric monitoring of data streams for changes in location and scale. *Technometrics*, 53(4), 379–389.

Sebastião, R., Gama, J., Rodrigues, P.P., and Bernardes, J. (2010). Monitoring incremental histogram distribution for change detection in data streams. In *Knowledge discovery from sensor data*, 25–42. Springer.

Sugiyama, M., Kanamori, T., Suzuki, T., Plessis, M., Liu, S., and Takeuchi, I. (2013). Density-difference estimation. *Neural Comput.*, 25(10), 2734–2775.

Wang, J., Tsang, W.W., and Marsaglia, G. (2003). Evaluating kolmogorov's distribution. *J. Stat. Softw.*, 8(18).