

Ensemble LSDD-based Change Detection Tests

Li Bu[†], Cesare Alippi*, Dongbin Zhao[†]

[†] The State Key Laboratory of Management and Control for Complex Systems
Institute of Automation, Chinese Academy of Sciences, Beijing, China

Email: bulipolly@gmail.com, dongbin.zhao@ia.ac.cn

* Politecnico di Milano, Milano, Italy and Università Svizzera Italiana, Lugano, Switzerland
Email: cesare.alippi@polimi.it

Abstract—The least squares density difference change detection test (LSDD-CDT) has proven to be an effective method in detecting concept drift by inspecting features derived from the discrepancy between two probability density functions (pdfs). The first pdf is associated with the concept drift free case, the second to the possible post change one. Interestingly, the method permits to control the ratio of false positives. This paper introduces and investigates the performance of a family of LSDD methods constructed by exploring different ensemble options applied to the basic CDT procedure. Experiments show that most of proposed methods are characterized by improved performance in change detection once compared with the direct ensemble-free counterpart.

I. INTRODUCTION

Ensemble learning, combining several diverse models to solve a given task, has been successfully used in many learning problems for its appealing properties [1]. Here, diversity among the components of the ensemble is proven to be a key issue [2], and in order to achieve diversity, the different models composing the ensemble can be generated by relying on different training subsets, configuration parameters, feature selection and even model architectures.

In order to build diverse base learners, e.g., base classifiers, Street [3] proposed to divide historical data into non-overlapped sequential chunks to train each classifier and adopted a replacement strategy to keep a fixed-size ensemble. Another method using different sizes of training windows to design the ensemble was proposed in [4]; the weights associated with learners were obtained according to historical performances. Different types of base learners were also commonly applied to ensure the diversity. In [5], three classifiers, i.e., the nearest neighbour classifier (1-NN), the perceptron with a fixed learning rate and an online linear discriminant classifier, were considered as base learners with constant update. Minku et al. analyzed the impact of diversity on different types of concept drifts [6] and proposed an ensemble approach with different diversity levels [7] which improved the learning performance.

The abovementioned methods are passive classifiers, i.e., the classifiers adapt online without the need of detecting a change. They do not consider the computational complexity or the memory requirement to be a constraint [8]. Active systems, coupled with change detection tests (CDT), are more sensitive

to the computational issue and update models only when a change is detected [9] [10]. In this scenario, the performance of a CDT is a key issue since a quick reaction relies on the promptness of the change detection. Many methods, like CICUSUM [9], H-ICI [11], and CPM [12], have been studied and used in different applications, whereas most of them operate on scalar streams and can't deal with true multi-dimensional problems. Sugiyama et al. [13] proposed a method that estimated the difference between two pdfs in multi-dimensional applications.

Few papers in the literature adopt the ensemble strategy to build a possibly high-performing change detection mechanism. Alippi et al. [14] introduced an ensemble of CPMs, and experiments showed the validity of the method even when residuals are not independent and identically distributed (i.i.d.). However, the study in [15] claimed that the ensemble method built by several Mann-Whitney (MW) tests [16] performed worse on i.i.d. data than the regular MW test by introducing more false positives.

In this paper, we propose several ensemble based change detection tests and investigate how different types of ensemble strategies influence the detection performance. The LSDD method [17] is used as base model, where large estimated values indicate significant changes and a threshold can be easily derived according to a predefined false positive (FP) rate. In the following, proposed ensemble methods are designed by diversifying w.r.t. the ensemble architecture or reference sets. In addition, an ensemble method based on LSDD proposed in [18] will also be compared here, which improves the detection performance with reduced false positives. It uses several reference sets during the testing phase, each of which is adapted with reservoir sampling.

The structure of the paper is as follows. Section II briefly introduces the LSDD method. Section III describes the proposed ensemble methods based on LSDD CDTs (LSDD-Ens). The results are given in section IV.

II. LSDD METHOD

Density-difference between two pdfs was firstly proposed in [13] to measure the least squares density difference:

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

This work was supported in part by the National Natural Science Foundation of China under Grants No.61273136, No. 61573353 and No.61533017.

where $x \in R^d$ is a real vector, and $p(x), q(x)$ are two pdfs generating the reference set Z_p and testing set Z_q separately. A Gaussian kernel model is used to represent $p(x) - q(x)$:

$$g(x, \Theta) = \sum_{i=1}^{2n} \theta_i \exp\left(-\frac{\|x - c_i\|_2^2}{2\sigma^2}\right), \quad \Theta = (\theta_1, \dots, \theta_i, \dots, \theta_{2n}), \quad (2)$$

where $(c_1, \dots, c_i, \dots, c_{2n}) = (x_{p,1}, \dots, x_{p,n}, x_{q,1}, \dots, x_{q,n})$ are d -dimensional kernel centers, n associated with pdf $p(x)$ and n with $q(x)$. Θ is a parameter vector, and σ the scale parameter $\sigma = \text{median}(\|x_i - x_j\|_2, 0 < i < j \leq 2n)$ [19]. The optimal parameter Θ^* is achieved by minimizing loss $J(\Theta)$:

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

where the $L2$ -regularizer is added to control over-fitting, and $\lambda > 0$. The optimization problem is then transformed to:

$$\begin{aligned} \hat{\Theta} &= \arg \min_{\Theta} (\Theta^T H \Theta - 2h^T \Theta + \lambda \Theta^T \Theta) \\ &= (H + \lambda I)^{-1} h, \end{aligned} \quad (4)$$

where H is a $2n \times 2n$ matrix, and h is the $2n \times 1$ vector:

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

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

$i, j = 1, \dots, 2n$. Since pdfs $p(x), q(x)$ are unknown, an empirical estimator \hat{h}_i is used instead:

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

Finally, $\hat{\Theta}$ becomes:

$$\hat{\Theta} = (H + \lambda I)^{-1} \hat{h}. \quad (8)$$

By replacing $p(x) - q(x)$ with $g(x, \hat{\Theta})$, the D^2 -distance can be estimated by two equivalent expressions $\hat{D}^2(p, q) \approx \hat{h}^T \hat{\Theta}$ and $\hat{D}^2(p, q) \approx \hat{\Theta}^T H \hat{\Theta}$, and in order to reduce the bias, we can consider [13] [17]:

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

Since parameter λ affects the smoothness of computed LSDD values, cross validation has been suggested to identify the optimal λ [13] when the two pdfs are different. Differently, a relative difference based method is proposed in [17] when $p(x) = q(x)$. We consider the latter one here since change detection works under the assumption that the training set

is in stationary, i.e., the two pdfs are identical. The relative difference is defined as:

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

A larger λ brings a smoother fitting, whereas it also introduces a larger difference between the true density difference and the estimated one. An appropriate λ can be found from several alternatives by controlling the relative difference RD according to a preset value.

III. LSDD-ENS

The basic LSDD method is designed to detect changes in datastreams by comparing the two pdfs from the theory point of view and two datasets from the operational point of view. The first dataset Z_p includes samples associated with pdf $p(x)$ and the Z_q with pdf $q(x)$ (the dataset is associated with the sliding window containing recently collected samples). Generally, the two subsets, working as the reference set and the testing set respectively, are assumed to be generated with different pdfs $p(x) \neq q(x)$. A LSDD CDT is then used to determine if the difference between the pdfs is significant, i.e., whether there is a change in the pdfs or not. However, the estimated \hat{D}^2 values strongly depend on the particular realization of samples and their distributions. In this case, neither a fixed constant nor a function associated with thresholds can be obtained.

A recent study [17] proposed to learn thresholds from the training set satisfying the stationary hypothesis. This method uses a bootstrap procedure to generate training subsets and computes LSDD values by comparing any two subsets; after fitting the distribution of obtained values with a Gamma distribution, a threshold T can be derived by estimating the upper tolerance limit with required FP rate μ [17] [18]

$$Pr(\hat{D}^2 > T) = \mu, \quad (9)$$

where the inequality holds with a confidence level γ . Once γ is preset, e.g., 0.99, T depends only on μ .

By introducing diversity in the ensemble in several ways, we propose a family of ensemble based LSDD CDTs (LSDD-Ens) to build possibly high-performing change detection mechanisms and investigate how they influence the detection performance. The first two ensemble methods are constructed by using different training subsets; two methods are designed by different configuration parameters λ on the same training set; another two methods build one model on the training set as in LSDD, and use M variable reference sets during the testing phase; the last two methods fix sets $Z_{p,i}(i = 1, \dots, M)$ during both the training and test phases, the first considers M components and the second only one. Details about the eight proposed ensemble methods are described in the sequel.

A. LSDD-Ens with Different Training Data

Since using different training sets to train individual models is the most popular method to generate ensemble, we firstly introduce two methods with different access to the training subsets.

The first method (*EnsBts*) generates M training subsets through a resampling technique to build a M -component ensemble. Each subset with size N_t is drawn randomly with replacement from the training set, and then used to train a LSDD model with the same procedure of LSDD-CDT [17]. For each model ($\hat{D}_i^2, i \in \{1, \dots, M\}$), the reference set $Z_{p,i}$ is fixed with n observations sampled from the training subset; then we obtain M models different with the training configuration. To complete the procedure, M thresholds, each of which is associated with one training subset and its corresponding reference set, are derived. Finally, we combine M detection results with majority voting: a change is detected only when the majority of tests signal a change.

A different ensemble method (*EnsSeg*) segments the training set into M non-overlapped subsets, and each model is trained on $M - 1$ subsets leaving one out, like cross validation. The procedure ensures that each training subset has at least N_t/M samples different from others, which guarantees the diversity level requested by the ensemble. The training and testing procedures of *EnsSeg* are the same as in *EnsBts* with fixed reference sets and majority voting.

B. LSDD-Ens with Different Parameters

As described in Section II, parameter λ affects the smoothness, the accuracy of density difference estimation, and the detection performance. Thus, in order to avoid an inappropriate λ with poor performance, we propose two ensemble methods whose components are trained with different values of λ .

The first method (*EnsParf*) uses a fixed reference set Z_p , and the second one (*EnsParUf*) adopts a variable reference set updated with reservoir sampling [20]. Since the diversity here relies on the values of λ , for each ensemble, we use the same reference set Z_p for all M components. Once Z_q is updated with new samples, M tests are designed on Z_p and Z_q with different λ ; the results are combined with majority voting.

C. LSDD-Ens with Variable Reference Sets

Detection performance is associated with the initially selected n samples when detecting changes, no matter the reference sets are fixed or not. In this case, we can consider several variable reference sets to reduce the possibility in ending up with unfortunate realizations of Z_p .

The training procedure is exactly the same as in [17] with one randomly sampled Z_p associated with threshold T . At the beginning of the test phase, M reference sets ($Z_{p,i}, i = 1, \dots, M$) are bootstrapped as the initial sets, and updated with reservoir sampling once new observations arrive. M LSDD values are obtained by comparing each $Z_{p,i}$ with the Z_q .

Here, two combination rules are considered to generate different ensembles. The first one (*EnsRefuV*) compares the M LSDD values with threshold T to obtain M detection results, and claims a change with the majority voting, which is as in [18]. The second one (*EnsRefuM*) averages the M LSDD values, and then the direct LSDD CDT is executed.

We remark here that only one model is built during the training phase, and the diversity relies on different reference sets when detecting changes.

D. LSDD-Ens with Fixed Reference Sets

In this scenario, we consider M fixed reference sets during both the training and testing phases, and introduce two associated ensemble methods. M sets $Z_{p,i} (i = 1, \dots, M)$ are randomly sampled from the training set and then keep fixed. The first method (*EnsReffV*) includes M components built on the training set, each corresponding to a fixed $Z_{p,i}$; detection results are combined with majority voting.

Differently, the second method (*EnsReffM*) designs only one model with one threshold T ; the M LSDD values associated with M reference sets are averaged and fed into the CDT. A change is detected only when the average exceeds T .

Different options can be summarized as follows:

- 1) Different training data
 - *EnsBts*: M training subsets generated with a bootstrap procedure;
 - *EnsSeg*: M training subsets generated by dividing training set into non-overlapped subsets;
- 2) Different training parameters
 - *EnsParf*: one fixed Z_p ;
 - *EnsParUf*: one variable Z_p according to reservoir sampling;
- 3) Different variable reference sets
 - *EnsRefuV*: M detection results and majority voting;
 - *EnsRefuM*: average the M LSDD values to get one LSDD;
- 4) Different fixed reference sets
 - *EnsReffV*: M components; majority voting;
 - *EnsReffM*: one model by averaging M LSDD values;

It should be noted that the updating strategy for the testing set Z_q , i.e., the sliding window, is the same in all components of ensembles. However, different updating strategies of the reference set introduce different models, which affect the detection performance.

IV. EXPERIMENTS

We provide a comprehensive comparison of different ensemble LSDD CDTs on different applications to investigate their properties. The basic LSDD-CDT method [18] is considered here as a reference test, being not based on ensemble and showed to be effective in detecting changes in multi-dimensional applications. The method updates Z_p with a reservoir sampling and uses a three-level thresholds mechanism to detect changes.

Five simulated applications are considered, whose samples are generated with different distributions and coupled with different changes. Each application has 4000 samples and a change is added starting at point 2001 until the end. Application D1 is representative of small changes, while applications D2-5 are well-known multidimensional benchmarks. In particular,

- Application D1 generates data according to a Normal distribution. Changes happen with distribution shifting from $N(0, 0.5)$ to $N(0.2, 0.5)$.

- Application D2 is three dimensional with data satisfying a multivariate normal distribution. The distribution shifts from $N([0,0,0],[0.5,0,0;0,0.5,0;0,0,0.5])$ to $N([0,0,0],[0.5,0.4,0.4;0.4,0.5,0.4;0.4,0.4,0.5])$.
- Application D3 is a two-class rotating mixture of Gaussians as proposed in [21] 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 at $\Sigma_1 = \Sigma_2 = [0.5, 0; 0, 0.5]$.
- Application D4 is a circle problem [22] where data satisfy $(x_1 - a)^2 + (x_2 - b)^2 \leq r^2$, and changes occur with radius r shifting from 0.2 to 0.3. $a = b = 0.5$, and variables x_1, x_2 are uniformly distributed in interval [0,1].
- Application D5 refers to a SineV problem [22] $x_2 \leq a \sin(bx_1 + c) + d$. $a = b = 1$, $c = 0$, and changes occur with d drifting from -5 to 4. Variables x_1 and x_2 are uniformly distributed in intervals [0,10] and [-10,10] respectively.

The configuration parameters are fixed as follows. The size of training set N_t is 1000, the confidence level γ is 0.99, the number of bootstraps to extract LSDD values m is 2000, and the number of ensemble components M is set to 5. The FP rate μ corresponding to threshold T is set to 0.1%, and the relative difference RD is set to 0.25. The values of λ for *EnsParf* and *EnsParUf* are {0.2, 0.5, 1, 1.5, 2}.

We consider four indexes to evaluate the performance of proposed LSDD-Ens methods:

- False positive rate (FP(%)): it counts the rate of the experiments where the test detects a change when there is no change.
- False negative rate (FN(%)): it counts the rate of undetected real changes.
- Delay (in samples): it measures the detection promptness in terms of the detection delay. We record a delay only when the change is exactly detected, and both the mean and the standard deviation (in parenthesis) of the delay values are computed.
- Computational time (CT in seconds): it shows the execution time taken to perform the tests. All the experiments are executed on the same platform (Intel(R) Xeon(R) X5650 @2.66GHz, 48G RAM). Both the mean and the standard deviation (in parenthesis) of execution time are computed.

A. KS Tests of Gamma Distribution

Even though the feasibility of using a Gamma distribution has been verified to fit the LSDD values extracted with bootstrap, the scenario could be different here since fixed reference sets are considered. We firstly conduct a Kolmogorov Smirnov (KS) test to show whether it is appropriate to use bootstrap and Gamma distributions or not.

The null hypothesis of a KS test is that the bootstrap-based LSDD values satisfy a Gamma distribution; the alternative hypothesis is that they do not satisfy the distribution. $KS = 1$ means the KS test rejects the null hypothesis at the 5% significance level, and we record the rate of non-rejections,

TABLE I
KS TESTS OF GAMMA DISTRIBUTIONS

	D1	D2	D3	D4	D5
<i>EnsBts</i>	86.6%	89.36%	74.12%	89.88%	85.52%
<i>EnsSeg</i>	89.04%	89.96%	75.12%	90.4%	83.36%
<i>EnsParf</i>	88.56%	89.52%	78.04%	88.92%	81.76%
<i>EnsParUf</i>	76.72%	75.76%	52.12%	73.72%	58.12%
<i>EnsRefuV</i>	76.4%	68.6%	52%	74.6%	59.6%
<i>EnsReffV</i>	88.28%	89.28%	75.32%	89.76%	83.88%
<i>EnsReffM</i>	0.4%	7.2%	0.8%	3.2%	1%

i.e., $KS = 0$, on 500 trials. The results of KS tests are shown in Table I.

Since in an ensemble, there might be more than one component, we conducted the KS test on each model and averaged the non-rejection rates; the training procedures of *EnsRefuV* and *EnsReffM* are the same with one randomly sampled Z_p and one estimated Gamma distribution, so that they share the same results of KS tests.

For the most methods, fitting the bootstrap-based distributions with a Gamma distribution is appropriate and convinced, since most of the tests claim non-rejections of the null hypothesis. However, they reject the null hypothesis of *EnsReffM* in the vast majority of tests. The main reason is that averaging the extracted LSDD values in *EnsReffM* changes the distribution so that they do not satisfy a Gamma distribution any more. Actually, according to the central limit theorem, when M becomes infinite, the averaged values should tend towards a Normal distribution.

B. Comparative Performance Study

A comprehensive comparison is taken on applications D1-D5 among the proposed ensemble based methods and the reference test *LSDD-CDT*. At each application, we compute the FP and FN rates, and record the averages and the standard variances of delay and computational time at 200 trials. The required three predefined FP rates of *LSDD-CDT* are $\mu_s = 2\%$, $\mu_w = 1\%$ and $\mu_c = 0.1\%$ which uses the same FP rate, i.e., 0.1%, to confirm a change. The change detection performance is shown in Table II.

EnsBts, *EnsSeg* and *EnsReffV* show improved performance compared to *LSDD-CDT*. In detail, FP and FN rates decrease, and are always lower than those associated with the *LSDD-CDT* method; moreover, ensemble CDTs show a smaller detection latency.

Detection results of *EnsParf* and *EnsParUf* are very similar to the *LSDD-CDT* ones; FP and FN rates and detection delay are comparable.

The FP rates of *EnsRefuV* and *EnsReffM* are the lowest among all applications, and lower than the reference ones. However, their FN rates increase a bit especially in application D1 where changes are very small. The reason being that the combined M reference sets ($Z_{p,i}, i = 1, \dots, M$) reduce the detection sensitivity and only significant changes can be detected.

TABLE II
PERFORMANCE COMPARISON ON DIFFERENT APPLICATIONS

	<i>LSDD-CDT</i>	<i>EnsBts</i>	<i>EnsSeg</i>	<i>EnsParf</i>	<i>EnsParUf</i>	<i>EnsRefuV</i>	<i>EnsRefuM</i>	<i>EnsReffV</i>	<i>EnsReffM</i>
D1	FP(%)	15.5	6	3	12	12.5	2	1	41
	FN(%)	25	2.5	1	14	36	35.5	38	0
	Delay	349.1 (409.45)	253.45 (248.82)	299.54 (356.46)	318.75 (380.16)	415.55 (436.88)	387.45 (413.75)	339.81 (375.19)	254.93 (246.64)
	CT	12 (5.54)	40.98 (10.43)	41.97 (10.61)	31.23 (12.63)	38.09 (16.07)	49.84 (21.72)	50.63 (21.79)	27.45 (9.05)
D2	FP(%)	15	9	8.5	17.5	10.5	3	2	38.5
	FN(%)	0	0	0	0	2	1	0.5	0
	Delay	101.93 (67)	86.61 (37.62)	87.35 (25.1)	99.48 (65.4)	122.47 (106.81)	108.48 (49.38)	112.97 (55.72)	84.68 (20.84)
	CT	11.34 (2.48)	51.45 (9.33)	51.6 (8.82)	38.61 (9.06)	40.86 (12.16)	45.69 (10.11)	46.47 (6.79)	39.78 (7.8)
D3	FP(%)	20.5	15	14.5	19	19	6	2	43.5
	FN(%)	0	0	0	0	0	0	0.5	0
	Delay	74.42 (16.81)	70.69 (12.76)	71.25 (12.67)	87.79 (129.08)	85.86 (34.24)	82.75 (17.05)	81.15 (13.16)	71.86 (13.21)
	CT	10 (2.15)	43.08 (9.01)	43.17 (8.87)	36.14 (9.85)	35.99 (9.06)	38.48 (6.26)	39.68 (6.28)	32.75 (9.04)
D4	FP(%)	14	8	9	15	11	1.5	2.5	42
	FN(%)	0	0	0	0	0	0	0	0
	Delay	62.08 (13.87)	60.5 (13.13)	60.96 (14.08)	59.76 (15.64)	66 (14.1)	69.36 (13.05)	69.2 (12.99)	61.13 (14.57)
	CT	9.17 (1.54)	37.6 (5.42)	37.1 (5.12)	27.26 (5.24)	27.77 (5.1)	32.38 (3.09)	32.3 (2.18)	26.79 (5.3)
D5	FP(%)	18	10.5	14	20	18	7.5	4.5	44
	FN(%)	0	0	0	0	0	0	0	0
	Delay	40.05 (7.95)	38.2 (9.05)	37.98 (8.37)	42.79 (11.94)	47.92 (11.94)	44.32 (7.58)	44.31 (7.16)	38.22 (8.43)
	CT	9.43 (1.73)	38.88 (5.96)	38.29 (6.39)	35.37 (8.32)	35.62 (8.47)	32.93 (4.62)	33.31 (4.23)	28.42 (7.15)

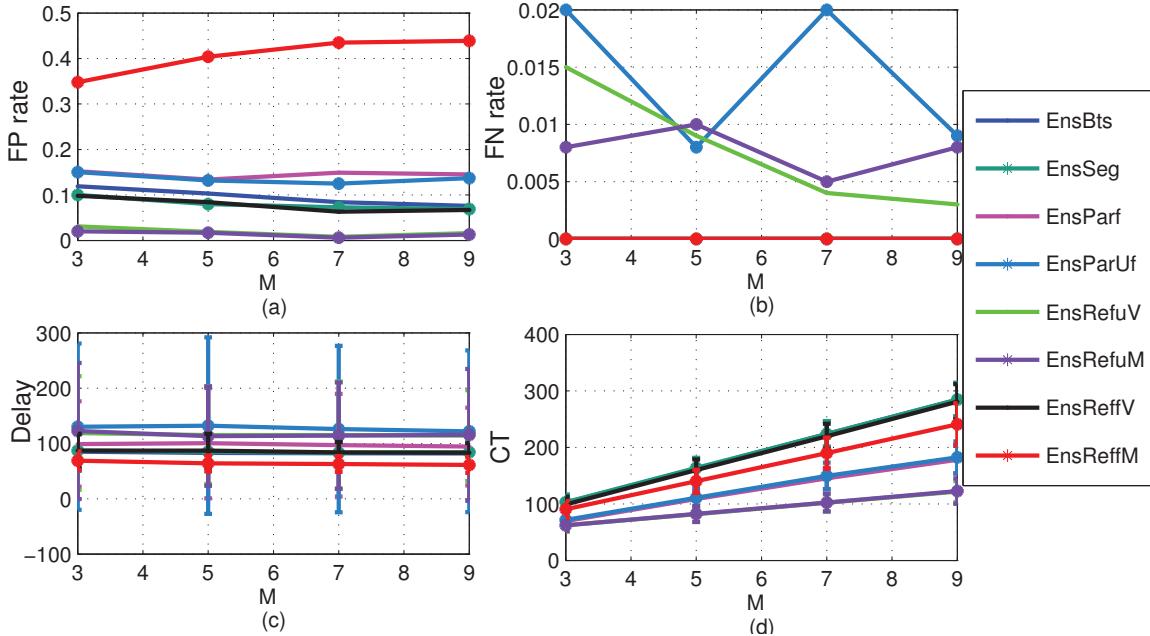


Fig. 1. Detection performance with different M

As expected, the detection performance of *EnsReffM* is the worst with too high FP rates. In fact, computing the average of LSDD values has changed the distribution. Thus, deriving thresholds with a Gamma distribution is not appropriate here.

In general, we comment that obtained thresholds are much smaller than the desired ones, which cause the high FP rates.

In order to evaluate how the number of ensemble components influences the detection performance, we operate

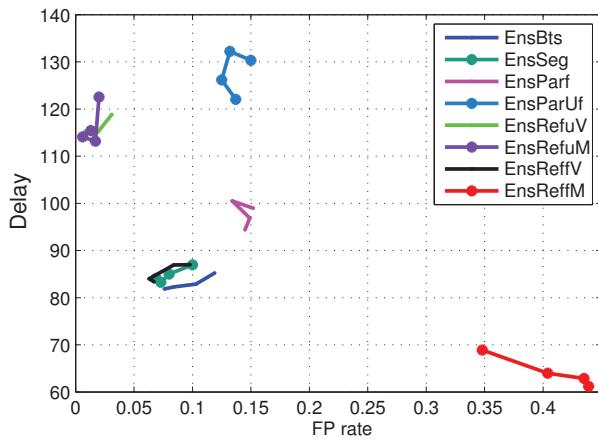


Fig. 2. FP rate vs. delay with different M

an experiment on application D2, which is a real multi-dimensional application. Since too large values of M will not change the performance trend, four values of M are considered as $\{3, 5, 7, 9\}$. Most of the experiment settings are the same as described above, and each experiment repeats 1000 times. Alternative values of λ for *EnsParf* and *EnsParUf* are $\{1, 0.5, 1.5, 0.2, 2, 0.1, 5, 0.05, 10\}$, and the first M values are used. The detection performances are shown in Fig.1. In these subplots, overlap happens and makes many of the lines invisible. However, the trend is clear and we will explain the results in detail.

It's obvious that with the increase of M , FP rates of most methods decrease as expected, and those of *EnsParf* and *EnsParUf* increase with M . However, the FP rates of *EnsReffM* increase, which further verifies the improper use of the Gamma distribution. Most of the FN rates stay at 0. From Fig.1(c), it seems that the increase of M shows no significant influence to the detection promptness. It's obvious that the computational time of Fig.1(d) increases linearly with M for all the methods. In particular, we can sort the CT from high to low as: *EnsSeg* \approx *EnsBts* \approx *EnsReffV* $>$ *EnsReffM* $>$ *EnsParUf* \approx *EnsParf* $>$ *EnsRefuV* \approx *EnsRefuM*, where the symbol \approx means that the associated methods is characterized by a similar CT and the former is slightly more time-consuming.

We then compare of the FP rate vs. delay since they are the most relevant indexes. Results are shown in Fig.2, which are consistent with those of table II. *EnsRefuV* and *EnsRefuM* detect changes with the lowest FP rates for arbitrary values of M , but with a larger delay. *EnsBts*, *EnsSeg* and *EnsReffV* show similar performance even with different M , and FP rates and delay are low. *EnsParf* and *EnsParUf* have similar FP rates, whereas the former seems better with much smaller delay than the latter.

V. CONCLUSIONS

In this paper, we propose a family of ensemble based change detection tests derived from the pdf-free LSDD CDT method. Comprehensive experiments show that ensemble methods,

when properly designed, improve the change detection performance. Most of the proposed ensembles outperform the reference test *LSDD-CDT* in detection accuracy and promptness, at an inevitable cost of higher computational time.

REFERENCES

- [1] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple classifier systems*. Springer, 2000, pp. 1–15.
- [2] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine learning*, vol. 51, no. 2, pp. 181–207, 2003.
- [3] W. N. Street and Y. Kim, "A streaming ensemble algorithm (sea) for large-scale classification," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 377–382.
- [4] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: An ensemble method for drifting concepts," *The Journal of Machine Learning Research*, vol. 8, pp. 2755–2790, 2007.
- [5] J. J. Rodríguez and L. I. Kuncheva, "Combining online classification approaches for changing environments," in *Structural, Syntactic, and Statistical Pattern Recognition*. Springer, 2008, pp. 520–529.
- [6] L. Minku, A. P. White, and X. Yao, "The impact of diversity on online ensemble learning in the presence of concept drift," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 730–742, 2010.
- [7] L. L. Minku and X. Yao, "Ddd: A new ensemble approach for dealing with concept drift," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 4, pp. 619–633, 2012.
- [8] C. Alippi, D. Liu, D. Zhao, and L. Bu, "Detecting and reacting to changes in sensing units: the active classifier case," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 3, pp. 353–362, 2014.
- [9] C. Alippi and M. Roveri, "Just-in-time adaptive classifiers-Part I: Detecting nonstationary changes," *IEEE Transactions on Neural Networks*, vol. 19, no. 7, pp. 1145–1153, 2008.
- [10] C. Alippi and M. Roveri, "Just-in-time adaptive classifiers-Part II: Designing the classifier," *IEEE Transactions on Neural Networks*, vol. 19, no. 12, pp. 2053–2064, 2008.
- [11] C. Alippi, G. Boracchi, and M. Roveri, "A hierarchical, nonparametric, sequential change-detection test," in *The 2011 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2011, pp. 2889–2896.
- [12] 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.
- [13] M. Sugiyama, T. Kanamori, T. Suzuki, M. Plessis, S. Liu, and I. Takeuchi, "Density-difference estimation," *Neural Computation*, vol. 25, no. 10, pp. 2734–2775, 2013.
- [14] C. Alippi, G. Boracchi, and M. Roveri, "Ensembles of change-point methods to estimate the change point in residual sequences," *Soft Computing*, vol. 17, no. 11, pp. 1971–1981, 2013.
- [15] J. Andersson, "Locating multiple change-points using a combination of methods," 2014.
- [16] A. Pettitt, "A non-parametric approach to the change-point problem," *Applied statistics*, pp. 126–135, 1979.
- [17] L. Bu, D. Zhao, and C. Alippi, "A pdf-free change detection test for data streams monitoring," in *IEEE Symposium Series on Computational Intelligence*, 2015.
- [18] L. Bu, C. Alippi, and D. Zhao, "A pdf-free change detection test based on density difference estimation," *IEEE transactions on neural networks and learning systems*, submitted for publication.
- [19] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. K. Sriperumbudur, "Optimal kernel choice for large-scale two-sample tests," in *Advances in neural information processing systems*, 2012, pp. 1205–1213.
- [20] J. S. Vitter, "Random sampling with a reservoir," *ACM Transactions on Mathematical Software (TOMS)*, vol. 11, no. 1, pp. 37–57, 1985.
- [21] G. Ditzler and R. Polikar, "Hellinger distance based drift detection for nonstationary environments," in *IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*. IEEE, 2011, pp. 41–48.
- [22] L. L. Minku, A. P. White, and X. Yao, "The impact of diversity on online ensemble learning in the presence of concept drift," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 730–742, 2010.