Information Fusion 45 (2019) 202–214

Contents lists available at ScienceDirect

Information Fusion

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

Combining univariate approaches for ensemble change detection in multivariate data

William J. Faithfull[*,a], Juan J. Rodríguez[b], Ludmila I. Kuncheva[a]

[a] School of Computer Science, Bangor University, Dean Street, Bangor, Gwynedd LL57 1UT, Wales,UK
[b] Escuela Politécnica Superior, University of Burgos, Avda. de Cantabria s/n, Burgos 09006, Spain

A B S T R A C T

Detecting change in multivariate data is a challenging problem, especially when class labels are not available. There is a large body of research on univariate change detection, notably in control charts developed originally for engineering applications. We evaluate univariate change detection approaches —including those in the MOA framework — built into ensembles where each member observes a feature in the input space of an unsupervised change detection problem. We present a comparison between the ensemble combinations and three established 'pure' multivariate approaches over 96 data sets, and a case study on the KDD Cup 1999 network intrusion detection dataset. We found that ensemble combination of univariate methods consistently outperformed multivariate methods on the four experimental metrics.

1. Introduction

Change detection is, at its simplest, the task of identifying data points that differ from those seen before. It is often deployed in a supervised or unsupervised context: monitoring the error rate of a learning algorithm which processes the target data, or directly monitoring the target data. In the second context, we do not have class labels with which to estimate an error rate. Unsupervised change detection in a single variable is the univariate case of the problem and has been extensively studied over more than half a century, yielding widely used approaches such as control charts, and specifically, the cumulative sum chart (CUSUM) [1,2]. There are a variety of univariate methods across the literature from several fields. Basseville and Nikiforov [3] published a monograph on detectors of abrupt change in 1993. There are extensive method reviews in the overlapping field of novelty detection, by Markou and Singh [4] and Pimentel et al. [5], and in outlier detection by Ben–Gal [6]. There are many approaches from the classification literature intended to monitor the error-rate of the incoming data and adapt a deployed classifier accordingly. The MOA (massive online analysis) framework [7,8] is a popular open source tool for data stream mining, providing a number of approaches for univariate change detection, all of which we evaluate in this work.

We take inspiration from our previous study [9] where we use classifier ensembles to detect concept change in unlabelled multivariate data. We propose an ensemble of univariate detectors (which could be called a 'subspace ensemble') as a means of adapting established univariate change detection methods to multivariate problems. Our hypothesis is that such an ensemble should be competitive or better than 'pure' unsupervised multivariate approaches. We contribute the following: 1. An evaluation of which established univariate change detection methods are well suited to subspace ensemble combination over 96 common datasets. 2. Whether subspace ensembles outperform three established multivariate change detection methods, especially in high dimensions. 3. A reproducible reinterpretation of the widely used KDD Cup 1999 [10] network intrusion detection dataset as a change detection problem.

When generalising unsupervised change detection to multiple dimensions, the challenges proliferate – in how many features should we expect to see change before signalling? Can we reasonably assume that all features and examples are independent? Multivariate approaches often assume that each example is drawn from a multivariate process [11–14]. Thus, we need not assume that the features are independent. Multivariate change detection attempts to model a multivariate process by means of a function to evaluate the fit of new data (an example or a batch) to that model. Some works monitor components independently (Tartatovsky et al. [15] and Evangelista et al. [16]), meaning that the approach is unable to respond to changes in the correlation of the components. Whether or not this is a disadvantage, depends upon the context of the change.

Change may have a different definition for different problems. For example, if we wish to be alerted when the value of a stock is falling, a sudden rise might be irrelevant. If using a control chart with upper and

* Corresponding author.
E-mail addresses: w.faithfull@bangor.ac.uk (W.J. Faithfull), jjrodriguez@ubu.es (J.J. Rodríguez), l.i.kuncheva@bangor.ac.uk (L.I. Kuncheva).

https://doi.org/10.1016/j.inffus.2018.02.003
Received 28 February 2017; Received in revised form 17 January 2018; Accepted 10 February 2018
Available online 13 February 2018
1566-2535/

lower limits, only monitoring the lower limit might considerably lower the false alarm rate. If the problem is well known then a heuristic can be applied, but if that is the case, there is most likely training data available for a supervised approach. Unsupervised approaches must be robust in the face of unknown context. The change we wish to detect could be abrupt or gradual. It could be a single change or repeating concepts. When we move into multiple dimensions, there is even more scope for contextual properties to stretch our assumptions. Change could manifest itself in a single feature, all features, or any number of features in-between. From the novelty detection literature, Evangelista et al. [16] conclude that unsupervised learning in subspaces of the data will typically outperform unsupervised learning that considers the data as a whole. In the course of this work, we investigate whether this assertion is reproducible.

The dimensionality of the input data presents a potential challenge. Allipi et al. [17] analyse the effect of an increasing data dimension d on change detectability for log-likelihood based multivariate change detection methods. They demonstrate that in the case of Gaussian random variables, change detectability is upper-bounded by a function that decays as $\frac{1}{d}$. Importantly, the loss in detectability arises from a linear relationship between the variance of the log-likelihood ratio and the data dimension. Evangelista et al. [16] propose that subspace ensembles are also a means to address the curse of dimensionality.

Multivariate detectors treat features as components of an underlying multivariate distribution [11]. We will term such detectors 'pure' multivariate detectors. For pure detectors to work well, the data dimensionality d should not be high, as Allipi et al. argued, and the data coming from the same concept should be available in an *i.i.d* sequence. This is rarely the case in practice. For example, Tartatovsky et al. [15] observe that the assumption that all examples are *i.i.d* is very restrictive in the domain of network intrusion detection.

The remainder of the paper is organised as follows. Section 2 covers the background and related work for this problem. Section 3 details the methods used, explains our combination mechanism, and overviews the experimental protocol. Our results are presented in Section 4, and our conclusions follow in Section 5.

2. Background & related work

Learning methods are frequently deployed in non-stationary environments, where the concepts may change unpredictably over time. Where class labels are immediately or eventually available, change detection methods can be required to monitor only a univariate error stream from a learner. When a change is detected in the error stream, we can retrain or adapt the model as required. However, when labels are not available, then we cannot use the error rate as a performance indicator. In this instance, a fully unsupervised approach must be taken.

Surveys by Gama et al. [18] and Ditzler et al. [19] discuss the distinction between real and virtual concept drift. Real concept drift is a change in the class conditional probabilities, i.e. the optimal decision boundary. Virtual concept drift refers to a change in the prior probabilities, or distribution of the data. Since in an unsupervised setting, we have no class labels to identify real concept drift, this work would conform to the latter definition. This particular problem formulation is closely related to the assumptions of statistical process control, novelty detection, and outlier detection, for which applications are usually unsupervised, and methods are expected to be applied directly to the domain data.

Most methods for multivariate change detection require two components: a means to estimate the distribution of the incoming data, and a test to evaluate whether new data points fit that model. Estimation of the streaming data distribution is commonly done by either clustering, or multivariate distribution modelling. Gaussian mixture models (GMM) are a popular parametric means to model a multivariate process for novelty detection, as in Zorriassatine et al. [12]. Tarassenko et al. [20] and Song et al. [21] use nonparametric Parzen windows (kernel density estimation) to approximate a model against which new data is compared. Dasu et al. [22] construct *kdq* trees to a similar effect. Krempl et al. [23] track the trajectories of online clustering, while Gaber and Yu [24] use the deviation in the clustering results to identify evolution of the data stream. Kuncheva [11] applies k means clustering to the input data and uses the cluster populations to approximate the distribution of the data.

Multivariate statistical tests for comparing distributions such, as Hotelling's t-squared test [25] need to be adapted into the sequential form over time windows of the data [11]. Bespoke statistics continue to be developed for this purpose [13,14]. Kuncheva [11] introduces a family of log-likelihood ratio detectors which use two time-windows of multivariate data to compute the probability that both are drawn from the same distribution. The observation that log-likelihood based detectors effectively reduce the input space to a univariate statistic can be further exploited, by monitoring that ratio with existing univariate methods [26].

Ensemble methods for monitoring evolving data streams is a growing area of interest within the change detection literature. There are recent surveys on the subject by Krawczyk et al. [27] and Gomes et al. [28]. The former observe that there has been relatively little research on the combination of drift detection methods. The publications that they review in this area [29,30] deal with the combination of detectors over univariate input data, in contrast to our own formulation. The latter work introduces a taxonomy for data stream ensemble learning methods, and demonstrates the diversity of available methods for ensemble combination. Du et al. [31] utilise an ensemble of change detectors in a supervised approach for a univariate error stream. Alippi et al. [32] introduce hierarchical change detection tests (HCDTs) combining a fast, sequential change detector with a slower, optionally-invoked offline change detector.

In the classification literature, ensemble change detection commonly refers to using these techniques to monitor the accuracy of classifiers in an ensemble, in order to decide when to retrain or replace a classifier [33–36]. Many of these established univariate methods for change detection are geared towards the supervised scenario which offers a discrete error stream [37,38]. The streaming ensemble algorithm (SEA) [39] was one of the first of many ensemble approaches for streaming supervised learning problems. However, instead of relying on a change detection, SEA creates an adaptive classifier which is robust to concept drift. Evangelista et al. [16] use a subspace ensemble of one-class support vector machine classifiers in the context of novelty detection. The input space is divided into 3 random subspaces, each monitored by a single ensemble member. Kuncheva [9] uses classifier ensembles to directly detect concept change in unlabeled data, sharing the same problem formulation as this work.

3. Change detection methods

The methods we evaluated are detailed in Tables 1 and 2. We chose to evaluate all the univariate detectors offered by MOA [7,8], an open source project for data stream analysis. Our experiment performs an unsupervised evaluation of all reference implementations of the `ChangeDetector` interface in the MOA package

`moa.classifiers.core.driftdetection` [1]

The interface contract implies the following basic methods to provide an input and subsequently check if change was detected:

```
public void input(double inputValue);
public boolean getChange();
```

All the univariate detectors are provided by MOA except CUSUM1, which is a CUSUM chart with upper and lower limits which was implemented in Java, and integrated into the experiment to serve as a

[1] https://github.com/Waikato/moa/tree/master/moa/src/main/java/moa/classifiers/core/driftdetection.

Table 1
Methods for change detection in univariate data.

Method	References	Category
SEED	[40]	Monitoring distributions
ADWIN	[8,41]	Monitoring distributions
SEQ1	[42]	Monitoring distributions
Page-Hinkley	[1,8]	Sequential analysis
CUSUM1	[1]	Sequential analysis
CUSUM2	[8]	Sequential analysis
GEOMA	[43,44]	Control chart
$HDDM_A$	[36]	Control chart
EDDM	[8,38]	Control chart
DDM	[8,37]	Control chart
EWMA	[8,43,44]	Control chart
$HDDM_W$	[36]	Control chart

Table 2
Methods for change detection in multivariate data.

Method	References	Category
SPLL	[11]	Monitoring distributions
Log-likelihood KL	[11]	Monitoring distributions
Log-likelihood hotelling	[11]	Monitoring distributions

baseline. We arrive at a final figure of 88 detectors, 3 of which are the multivariate approaches listed in Table 2, and the remaining 85 are ensembles of the univariate approaches with varying thresholds. The experimental details will be given in Section 3.2. A full list of the 96 datasets and their characteristics can be found in Table 4. Our metrics for evaluation and our experimental protocol are addressed in Section 3.3. Finally, we discuss the case study in Section 3.4.

3.1. Overview of the methods

The univariate detectors are listed in Table 1, with their accompanying publications. We categorise the methods based on the change detection taxonomy presented in Gama et al. [18]. What follows is a high-level overview of the theory behind each category of methods along with an abridged description of each detector. More details for each detector can be found in the accompanying publications in Table 1. The source code for each detector is available for inspection in the MOA repository.

3.1.1. Sequential analysis

Sequential analysis methods have much in common with the sequential probability ratio test (SPRT) [2]. Consider a sequence of examples $X = [x_1, ...,x_N]$. The null hypothesis H_0 is that X is generated from a given distribution $p_0(x)$, and the alternative hypothesis H_1 is that X is generated from another (known) distribution $p_1(x)$. The logarithm of the likelihood ratio for the two distributions is calculated as

$$\Lambda_N = \sum_{i=1}^{N} \log \frac{p_1(x_i)}{p_0(x_i)}$$

Two thresholds, α and β are defined depending on the target error rates. If $\Lambda_N < \alpha$, H_0 is accepted, else if $\Lambda_N > \beta$, H_1 is accepted. In the case where $\alpha <= \Lambda_N <= \beta$, the decision is postponed, the next example in the stream, x_{N+1}, is added to the set, and Λ_{N+1} is calculated and compared with the thresholds. Cumulative sum (CUSUM) [1] is a sequential analysis technique based on the same principle. The test is widely used for detecting significant change in the mean of input data. Starting with an upper cumulative sum statistic $g\bigtriangleup_0 = 0$, CUSUM updates $g\bigtriangleup$ for each subsequent example as

$$g\bigtriangleup_i = \max(0, g\bigtriangleup_{i-1} + (x_i - \delta))$$

where δ is the magnitude of acceptable change. Change is signalled when $g\bigtriangleup_i > \lambda$, where λ is a fixed threshold. If we wish to detect both positive and negative shifts in the mean, we can also compute and threshold the lower sum as

$$g\bigtriangledown_i = \min(0, g\bigtriangledown_{i-1} - (x_i - \delta))$$

The Page–Hinkley test [1] is derived from CUSUM, and adapted to detect an abrupt change in the average of a Gaussian process [18,45].

3.1.2. Control charts

Control charts[2] are a category of methods that are based upon statistical process control (SPC). In SPC, the modus operandi is to consider the problem as a known statistical process, and monitor its evolution. Assume that we monitor classification error. This error can be interpreted as a Bernoulli random variable with probability of "success" (where error occurs) p. The probability is unknown at the start of the monitoring, and is re-estimated with every new example as the proportion of errors encountered thus far. At example i, we have a binomial random variable with estimated probability p_i and standard deviation $\sigma_i = \sqrt{p_i(1-p_i)/i}$. One way to use this estimate is described below [18,37]:

1. Denote the (binary) streaming examples as $x_1, x_2, ...$ To keep a running score of the minimum p, start with estimate $p_{\min} = 1$, and $\sigma_{\min} = 0$. Initialise the stream counter $i \leftarrow 1$.
2. Observe x_i. Calculate p_i and σ_i. For an error and a standard deviation (p_i, σ_i) at example x_i, the method follows a set of rules to place itself into one of three possible states: in control, warning, and out of control. Under the commonly used confidence levels of 95% and 99%, the rules are:
   - If $p_i + \sigma_i < p_{\min} + 2\sigma_{\min}$, then the process is deemed to be in control.
   - If $p_i + \sigma_i \geq p_{\min} + 3\sigma_{\min}$, then the process is deemed to be out of control.
   - If $p_{\min} + 2\sigma_{\min} \leq p_i + \sigma_i < p_{\min} + 3\sigma_{\min}$, then this is considered to be the warning state.
3. If $p_i + \sigma_i < p_{\min} + \sigma_{\min}$, re-assign the minimum values: $p_{\min} \leftarrow p_i$ and $\sigma_{\min} \leftarrow \sigma_i$.
4. $i \leftarrow i + 1$. Continue from 2.

The geometric moving average chart (GEOMMA), introduced by Roberts [44], assigns weights to each observation such that the weight of older observations decreases in geometric progression. This biases the method towards newer observations, improving the adaptability. Exponentially weighted moving average (EWMA) charts are a progression of this approach such that the rate of weight decay is continuous and can be tuned.

The EWMA charts used by Ross et al. [43] expect the initial distribution to have known parameters, which is a restrictive assumption in the area of change detection. To address this limitation, the initial distribution is approximated in advance through regression of the distributional parameters to achieve a desired average running length (ARL).

Drift detection method (DDM) [37] is designed to monitor classification error using a control chart construction. It assumes that the error rate will decrease while the underlying distribution is stationary.

Similarly, the early drift detection method (EDDM) [38] is an extension of DDM which takes into account the time distance between errors as opposed to considering only the magnitude of the difference,

[2] A number of the control chart methods in MOA are intended for supervised predictive error monitoring rather than continuous data., however they accept continuous data by virtue of the `ChangeDetector` interface. While their assumptions are violated by the unsupervised experiment, we include their results for demonstrative purposes as MOA does not make a distinction. The Page–Hinkley detector might be expected to perform better on a prequential error stream [46], but retains valid assumptions for unsupervised features.

which is aimed at improving the performance of the detector on gradual change. HDDM$_A$ and HDDM$_W$ are extensions which remove assumptions relating the to probability density functions of the error of the learner. Instead, they assume that the input is an independent and bounded random variable, and use Hoeffding's inequality to compute the bounds [36].

3.1.3. Monitoring two distributions

The methods in this category monitor the distributions of two windows of data. The basic construction involves a reference window composed of old data, and a detection window composed of new data. This can be achieved with a static reference window and a sliding detection window, or a sliding pair of windows over consecutive observations. The old and new windows can be compared with statistical tests, with the null hypothesis being that both windows are drawn from the same distribution.

For fixed-sized windows, their sizes need to be decided *a priori*, which poses a problem. A small-sized window discards old examples swiftly, best representing the current state, but it also makes the method vulnerable to outliers. Conversely, a large-sized window provides more stable estimates of the probabilities and other variables of interest, but takes longer to pick up a change. In order to address this selection problem, there are a number of approaches for growing and shrinking sliding windows on the fly [41,47,48].

A widely-used approach of this type is adaptive windowing (ADWIN) by Bifet and Gavaldà [41]. It keeps a variable-length window of recently seen examples, and a fixed-size reference window. For the variable size window, ADWIN keeps the longest possible window within which there has been no statistically significant change. In its formulation as a change detector, change is signalled when the difference of the averages of the windows exceeds a computed threshold. When this threshold is reached, the reference window is emptied, and replaced by the variable length window, which is then regrown from subsequent observations. The SEQ1 algorithm [42] is an evolution of the ADWIN approach with a lower computational complexity. Cut-points are computed differently – where ADWIN makes multiple passes through the window to compute candidate cut-points, SEQ1 only examines the boundary between the latest and previous batch of elements. Secondly, the means of data segments are estimated through random sampling instead of exponential histograms. Finally, the authors employ the Bernstein bound instead of the Hoeffding bound to establish whether two sub-windows are drawn from the same population because the Hoeffding bound was deemed to be overly conservative.

In the SEED algorithm by Huang et al. [40], the data comes in blocks of a fixed size, so the candidate change points are the block's starting and ending points. Adjacent blocks are examined and grouped together if they are deemed sufficiently similar. This operation, termed 'block compression', removes candidate change points which have a lower probability of being true change points. Pooling blocks together amounts to obtaining larger windows, which in turn, ensures more stable estimates of the probabilities of interest compared to estimates from the original blocks. Drift detection is subsequently carried out by analysing possible splits between the newly-formed blocks.

3.1.4. Multivariate change detectors

Consider a random vector $\mathbf{x}$

$$\mathbf{x} = [x_1, x_2, ..., x_n]^T \in \mathbb{R}^n,$$

drawn from a continuous stream

$$\mathbf{x}_i, \mathbf{x}_{i+1}, \ ..., \mathbf{x}_N ...$$

We assume that $\mathbf{x}$ are drawn from a probability distribution $p_0(\mathbf{x})$ up to a certain point c in the stream, and from a different distribution thereafter. The objective is to find the change point c. We can estimate p_0 from the incoming examples and compute the likelihood $\mathscr{L}(\mathbf{x}|p_0)$ for subsequent examples. A successful detection algorithm will be able to identify c by a decrease of the likelihood of the examples arriving after c. To estimate and compare the likelihoods before and after a candidate point, the data is partitioned into a pair of adjacent sliding time-windows of examples, W_1 and W_2.

The Hotelling detector uses the multivariate T^2 test for equal means, and assumes equal covariance matrices of W_1 and W_2. Therefore, if the change of the distribution comes from change in the variances or covariances in the multidimensional space of the data, the test will be powerless.

As an alternative, we used a non-parametric change detector based on the Kullback–Leibler divergence (KL). To this end, the data in W_1 is clustered using k-means into K clusters, $C = \{C_1, ..., C_K\}$. A discrete distribution P is defined on C, where each cluster is given a probability equal to the proportion of examples it holds. The examples in W_2 are labelled in the K clusters by the nearest cluster centroid. The proportions of examples labelled in the respective cluster define the distribution Q over C, this time derived from the data in W_2. If the two distributions were identical, the KL divergence will be close to 0, and if they are very different, it will be close to 1. The success on this detector depends on a wise choice of the number of clusters K relative to the window sizes and the space dimensionality n. A smaller number of clusters ensures that there are enough points in each cluster to allow for reasonable estimates of the probability mass function. On the other hand, a larger number of clusters allows for better fidelity in approximating the distributions.

Finally, we include in the experiment the semi-parametric log-likelihood detector (SPLL) [11] as a compromise between the parametric detector (Hotelling) and non-parametric detector (KL). SPLL, like KL, applies k-means clustering to W_1 into K clusters. However, rather than approximating a discrete distribution, the criterion function of SPLL is derived assuming that we have fitted a Gaussian mixture with equal mixing proportion and common covariance matrix for the K clusters. The first part of the statistic of the SPLL detector is proportional to the mean of the squared Mahalanobis distances between each example in W_2 and its nearest cluster centroid. The calculation is repeated symmetrically by clustering first W_2, and then assigning labels to the examples in W_1. This gives the second part of the SPLL statistic. These two parts are subsequently averaged.[3]

3.2. Ensemble combination of univariate detectors

In order to evaluate univariate approaches on multivariate data, we adopted an ensemble combination strategy whereby each member monitors a single feature of the input space. This approach is analogous to using a subspaces ensemble method with a subspace size of 1, with as many subspaces and detectors as the dimensionality of the input space. Using subspaces with a size greater than 1, as in Evangelista et al. [16], would require combination of multivariate approaches. Fig. 1 shows an illustration of the ensemble combination scheme. In this set of experiments, the decisions are combined by a simple voting scheme with a variable threshold. Our naming convention for a single ensemble is as follows:

$$\text{DETECTOR - AGREEMENT THRESHOLD} \tag{1}$$

For example, ADWIN-30 refers to an ensemble of univariate ADWIN detectors, which requires 30% agreement at any given point to signal change. The multivariate detectors will simply be referred to as, KL, SPLL and Hotelling, as they are not ensembles.[4]

[3] MATLAB code is available at https://github.com/LucyKuncheva/Change-detection .

[4] The ensemble of multivariate detectors is a special case, because, unlike the ensembles of univariate detectors, it consists of only three detectors. In this case, the number of members does not scale with the number of features. As such, there is no benefit in having a scale of agreement thresholds when there are only ever 3 ensemble members. We chose 50% as a simple majority out of 3.

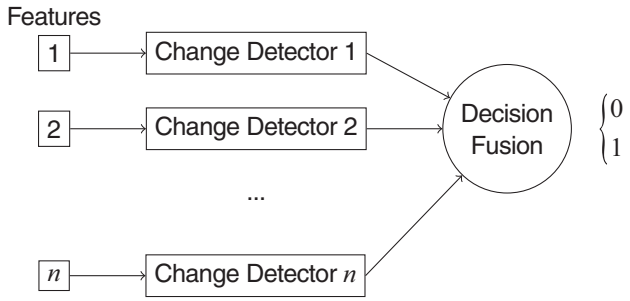

Fig. 1. An illustration of the ensemble combination scheme. All change detectors are of the same type, but each monitors a different feature.

Diversity is an important consideration when building an ensemble, because it implies that the members will make different mistakes [49,50] and there have been several analyses of ensemble diversity in evolving data streams [28,51]. However, unlike in these works, our ensembles consist of identical detectors. Diversity is introduced through the differing input to each detector. On a related note, there will be redundant features in the datasets, which will effect ensemble performance. Ideally this would be addressed through a feature extraction step, but such a measure is both difficult to generalise across datasets and outside the scope of this paper. As our ensembles are created with identical members, no one type of detector can gain an advantage in the results due to drawing many redundant features by chance.

3.3. Experimental protocol

The main experiment of this paper evaluates our multivariate change detection methods across the 96 datasets in Table 4. We evaluate the 3 multivariate detectors – SPLL, KL and Hotelling, an ensemble of these multivariate detectors, and 84 feature-wise ensembles of the univariate detectors with varying agreement thresholds, making a total of 88 detectors. A breakdown of the methods is presented in Table 3.

We note that when the thresholds in Table 3 are utilised on particularly small ensembles, the lower thresholds will become logically equivalent. For example, in ensembles with fewer than 20 members, the 5% and 1% thresholds will make the same decisions ($20 \times 0.5 = 1$). Since 43.33% of the datasets have more than 20 features, the difference in results between these lower thresholds will depend upon the larger datasets.

All the methods were evaluated against three rates of change: Abrupt, Gradual 100 and Gradual 300, for which we recorded separate sets of results. Algorithm 1 is a simplified pseudocode representation of the experiment. For each leg of the experiment, each detector is evaluated 100 times for each dataset. On each of these runs, we choose a random subset of the classes, and take this subset to represent distribution p_0 (before the change). The subset with the remaining classes is taken to represent distribution p_1 (after the change). Points are then sampled randomly, with replacement, from the p_0 and p_1 sets – 500 examples in the abrupt case, 600 and 800 respectively in the gradual cases. Denote these samples by S_1 and S_2, respectively. We add a small random value to each example, scaled by the standard deviation of the data, to avoid examples that are exact replicas. In the abrupt case, S_1 and S_2 are concatenated to create a 1000-example test sample with i.i.d stream from index 1 to 500, coming from p_0, followed by an abrupt change at index 500 to another i.i.d. stream of examples coming from p_1. To emulate gradual change over 100 examples, we take S_1 and S_2 as before, but do not concatenate them. At index 500, we sample with increasing frequency from S_2. The chance of an example coming from S_1 increases linearly from 1% at index 501 to 100% at index 600. Note that the class subsets for sampling S_1 and S_2 were chosen randomly for each of the 100 runs of the experiment.

As the chosen datasets are not originally intended as streaming data, our experiment uses the concept that the separable characteristics of each class are woven throughout the features. Therefore some changes will be easier to detect than others, introducing variety in our test data. Even if the sample size is insufficient to detect changes in a given dataset, this does not compromise experimental integrity because every detector faces the same challenge. A detector which performs well on average has negotiated a diverse range of class separabilities.

Datasets with fewer than 1000 examples will be oversampled in this experiment, but we found no relationship between the oversampling percentage of a dataset and our results. Even if this were to hinder or benefit the task at hand, the challenge is the same for every detector.

We measure the following characteristics for each method, averaged over the 100 runs each, for abrupt and gradual change on each dataset:

ARL Average running length: The average number of contiguous observations for which the detector did not signal change.
TTD Time to detection: The average number of observations between a change occurring and the detector signalling.
NFA The percentage of runs for which the detector did not issue a false alarm.
MDR The percentage of runs for which the detector did not signal after a true change.

Based on these characteristics, a good method should maximise *ARL* and *NFA*, and minimise *TTD* and *MDR*.

Fig. 2 is the archetype of our result figures. It plots *TTD* versus *ARL* for the detection methods. The grey dots correspond to ensemble methods, and the highlighted black dots correspond to the individual detectors (Hotelling, KL, and SPLL). The ideal detector will have $ARL = \infty$ (500 in our experiment, meaning that no false detection has been made before the true change happened), and $TTD = 0$. This detector occupies the bottom right corner of the plot. Dots which are close to this corner are indicative of good detectors.

The two trivial detectors lie at the two ends of the diagonal plotted in the figure. A detector which always signals change has $ARL = 0$ and $TTD = 0$, while detector which never signals change has $ARL = 500$ and $TTD = 500$. A detector which signals change at random will have its corresponding point on the same diagonal. The exact position on the diagonal will depend on the probability of signalling a change (unrelated to actual change). Denote this probability by p. Then *ARL* is the expectation of a random variable X with a geometric distribution (X is the number of Bernoulli trials needed to get one success, with

Table 3
The ensembles and detectors evaluated in the experiment.

Ensemble	Agreement thresholds	Count
SEED	1, 5, 10, 20, 30, 40, 50	7
ADWIN	1, 5, 10, 20, 30, 40, 50	7
SEQ1	1, 5, 10, 20, 30, 40, 50	7
PH	1, 5, 10, 20, 30, 40, 50	7
CUSUM1	1, 5, 10, 20, 30, 40, 50	7
CUSUM2	1, 5, 10, 20, 30, 40, 50	7
GEOMMA	1, 5, 10, 20, 30, 40, 50	7
$HDDM_A$	1, 5, 10, 20, 30, 40, 50	7
EDDM	1, 5, 10, 20, 30, 40, 50	7
DDM	1, 5, 10, 20, 30, 40, 50	7
EWMA	1, 5, 10, 20, 30, 40, 50	7
$HDDM_W$	1, 5, 10, 20, 30, 40, 50	7
MV	50	1
		Total 85
	Multivariate detector	**Count**
	SPLL	1
	KL	1
	Hotelling	1
		Total 3

Table 4
The 96 datasets used in the main experiment. N is examples, n is features and c is classes.

Dataset	N	n	c	Dataset	N	n	c
Abalone	4177	8	3	Molec-biol-splice	3190	60	3
Acute-inflammation	120	6	2	Monks-1	556	6	2
Acute-nephritis	120	6	2	Monks-2	601	6	2
Adult	48842	14	2	Monks-3	554	6	2
Annealing	850	31	3	Mushroom	8124	21	2
Arrhythmia	295	262	2	Musk-1	476	166	2
Balance-scale	576	4	2	Musk-2	6598	166	2
Bank	4521	16	2	Nursery	12958	8	4
Blood	748	4	2	oocytes_merluccius_nucleus_4d	1022	41	2
Breast-cancer	286	9	2	oocytes_merluccius_states_2f	1022	25	3
Breast-cancer-wisc	699	9	2	oocytes_trisopterus_nucleus_2f	912	25	2
Breast-cancer-wisc-diag	569	30	2	oocytes_trisopterus_states_5b	898	32	2
Car	1728	6	4	Optical	5620	62	10
Cardiotocography-10clases	2126	21	10	Ozone	2536	72	2
Cardiotocography-3clases	2126	21	3	pPage-blocks	5445	10	4
Chess-krvk	28029	6	17	Pendigits	10992	16	10
Chess-krvkp	3196	36	2	Pima	768	8	2
Congressional-voting	435	16	2	Planning	182	12	2
Conn-bench-sonar-mines-rocks	208	60	2	Ringnorm	7400	20	2
Conn-bench-vowel-deterding	990	11	11	Seeds	210	7	3
Connect-4	67557	42	2	Semeion	1593	256	10
Contrac	1473	9	3	Soybean	362	35	4
Credit-approval	690	15	2	Spambase	4601	57	2
Cylinder-bands	512	35	2	Spect	265	22	2
Dermatology	297	34	4	Spectf	267	44	2
Ecoli	272	7	3	Statlog-australian-credit	690	14	2
Energy-y1	768	8	3	Statlog-german-credit	1000	24	2
Energy-y2	768	8	3	Statlog-heart	270	13	2
Glass	146	9	2	Statlog-image	2310	18	7
Haberman-survival	306	3	2	Statlog-landsat	6435	36	6
Hayes-roth	129	3	2	Statlog-shuttle	57977	9	5
Heart-cleveland	219	13	2	Statlog-vehicle	846	18	4
heart-hungarian	294	12	2	steel-plates	1941	27	7
heart-va	107	12	2	synthetic-control	600	60	6
hill-valley	1212	100	2	teaching	102	5	2
horse-colic	368	25	2	thyroid	7200	21	3
ilpd-indian-liver	583	9	2	tic-tac-toe	958	9	2
image-segmentation	2310	18	7	titanic	2201	3	2
ionosphere	351	33	2	twonorm	7400	20	2
iris	150	4	3	vertebral-column-2clases	310	6	2
led-display	1000	7	10	vertebral-column-3clases	310	6	3
letter	20000	16	26	wall-following	5456	24	4
low-res-spect	469	100	3	waveform	5000	21	3
lymphography	142	18	2	waveform-noise	5000	40	3
magic	19020	10	2	wine	130	13	2
mammographic	961	5	2	wine-quality-red	1571	11	4
miniboone	130064	50	2	wine-quality-white	4873	11	5
molec-biol-promoter	106	57	2	yeast	1350	8	5

probability of success p), that is $ARL = \frac{1-p}{p}$. The time to detection, TTD, amounts to the same quantity because it is also the expected number of trials to the first success, with the same probability of success p. Thus the diagonal $ARL = TTD$ is a baseline for comparing change detectors. A detector whose point lies above the diagonal is inadequate; it detects change when there is none, and fails to detect an existing change. We follow the same archetype for visualisation of the MDR/NFA space. We plot MDR against 1-NFA for these figures in order to maintain the same visual orientation for performance. Therefore the ideal detector in this space is also at point (1, 0), i.e., all changes were detected, and there were no false alarms.

3.4. A case study

In addition to the main experiment, we conducted a practical case study on a network intrusion detection dataset. We chose the popular KDD Cup 1999 intrusion detection dataset, which is available from the UCI machine learning repository [10]. With a network intrusion dataset, the change context is more likely to be longer-lived change from one concept to another, which could be either abrupt or gradual. The dataset consists of 4,900,000 examples and 42 features extracted from seven-weeks of TCP dump data from network traffic on a U.S. Air Force LAN. During the seven weeks, the network was deliberately peppered with attacks which fall into four main categories.

- Denial of service (DOS): An attacker overwhelms computing resources in order to deny access to them.
- Remote to login (R2L): Attempts at unauthorised access from a remote machine, such as guessing passwords.
- Unauthorized to root (U2R): Unauthorised access to local superuser privileges, through a buffer overflow attack, for example.
- Probing: surveillance and investigation of weaknesses, such as port scanning.

Of these categories, there are 24 specific attack concepts, or 24 classes. This dataset is most commonly interpreted as a classification task. Viewed as such, it offers some interesting challenges in its deficiencies. For example, there is 75% and 78% redundancy in duplicated records across the training and testing set respectively [52]. This can serve to bias learning algorithms toward frequent records. It also has

```
for dataset in datasets do
    for i = 1, . . ., 100 do
        Choose a random subset of the classes as p_0;
        if abrupt then
            Sample 500 examples as S_1 from p_0;
        else if gradual 100 then
            Sample 600 examples as S_1 from p_0;
        else
            Sample 800 examples as S_1 from p_0;
        end
        Sample 500 examples as S_2 from the remaining classes;
        Concatenate subsets into 'abrupt' and 'gradual' test data;
        for detector in detectors do
            Evaluate abrupt;
            Evaluate gradual 100;
            Evaluate gradual 300;
        end
    end
    Store average abrupt metrics;
    Store average gradual 100 metrics;
    Store average gradual 300 metrics;
end
```

Algorithm 1. Experimental procedure.

very imbalanced classes, with the *smurf* and *neptune* DoS attacks constituting 71% of the data points; more than the 'normal' class. We offer an interpretation of this data as a change detection task.

We evaluated the methods on the testing dataset. Since the data is sequential, we pass observations in order, one-by-one to each of the detectors. The objective in our experiment was for the detectors to identify the concept boundaries. When the concept changes from one class to another, we record whether this change point was detected. With this scheme, if we are experiencing a long-lived concept such as a denial of service attack then after a sufficient number of examples of the same concept, we would expect the change detection methods to also detect the changepoint back to the normal class, or to another attack.

One challenge for the change detectors in this interpretation is that some concepts may be very short-lived, that is, the change in the distribution is a 'blip', involving only a few observations, after which the distribution reverts back to the original one. Such blips may be too short to allow for detection by any method which is not looking for isolated outliers.

4. Results and discussion

Fig. 3 visualises the ARL/TTD space for abrupt and gradual change type by the categories in the taxonomy by Gama et al. [18]. Each plot contains all 96 points (one for each data set) of the 88 change detection methods. Empirically, there is a clear and visible distinction between the methods in the control chart category, which performed, on average, worse than chance, and those in the other two categories. Table 5 confirms that Sequential Analysis and Monitoring distribution methods were much more likely to exhibit a high ARL. Furthermore, distribution monitoring methods exhibited considerably lower TTD whilst being competitive on ARL with Sequential Analysis methods. Observe the two distinct clusters in the ARL/TTD space for this category (the triangle marker), and the relative sparsity in-between. We suspect that this is the effect of gradual change on the TTD statistic. This is visible between the figures, where we observe that, in the gradual change experiment, those methods with a high ARL and low TTD struggle to better a TTD of 50, which is the halfway point of introducing the gradual change. Those methods with an already low ARL do not move significantly in the TTD axis between experiments. We suspect that this is because a low ARL implies an over-eager detector, which in turn increases the probability that a valid detection is due to random chance rather than a response to observation of the data.

The bottom two charts in Fig. 3 visualise the NFA/MDR space for the aforementioned categories. Interestingly, we see a very similar effect for control chart methods. To understand why the performance of this category is so poor, we must consider the assumptions of the detectors. This experiment presented the data points directly to the change detection methods in the ensemble. Specifically, this category contains EDDM, HDDM_A and HDDM_W, all of which share a common ancestor in DDM. Whilst the MOA interface for change detectors accepts 64 bit floating point numbers, these methods were not intended for continuous-valued data. As we mention in Section 3.1.2, DDM assumes the Binomial distribution. It also assumes that the monitored value (e.g., error rate of a classifier) will decrease while the underlying distribution is stationary. The derived methods also share this assumption, which is fundamentally violated by the nature of the data presented to them in this experiment.

The top 20 performers averaged over abrupt and gradual change are summarised in the left half of Table 6. The performers were ranked by minimum euclidean distance to the ideal points in the ARL/TTD and NFA/MDR spaces, (500, 0) and (1, 0).

The results for each individual method are summarised in the ARL/TTD space in Fig. 4, and in the NFA/MDR space in Fig. 5. In the ARL/TTD space, the SEED and ADWIN detectors were the best performers, with Page Hinkley, CUSUM2 and SEQ1 showing competitive patterns. The multivariate detectors exhibited a large standard deviation, suggesting that their performance is related to the suitability of the data – an observation which would appear to lend further credence to the conclusions of Allipi et al. [17], as well as our own hypothesis. In the NFA/MDR space, the winners are the low quorum ensembles of the SEED and ADWIN detectors. In fact, all the ensembles outside of the control chart category performed favorably compared to the multivariate detectors. Observing the curves of the SEED, ADWIN, Page Hinkley, CUSUM1, CUSUM2 and SEQ1 detectors across both sets of

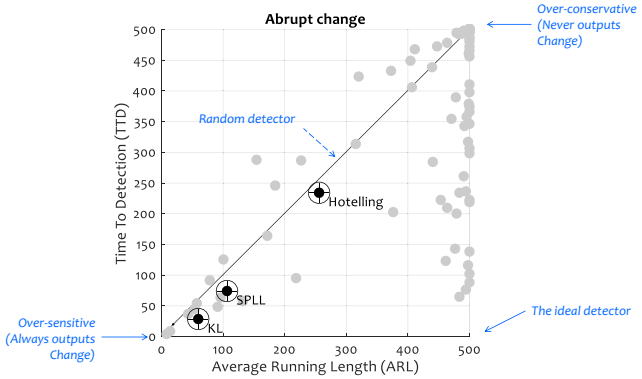

Fig. 2. Scatterplot of the 88 detector methods in the space (*ARL, TTD*) for the Abrupt-change part of the experiment. The three individual detectors are highlighted.

metrics, we see that the ideal agreement threshold is a case-by-case problem. The ADWIN ensemble improves almost linearly as we reduce the agreement threshold, suggesting that the optimum scheme is one whereby any member of the ensemble has absolute authority to signal a change. With other ensembles such as SEED and SEQ1, the 1% threshold is beyond the optimal, with the best ensembles having thresholds of 5% and 10%, respectively in the NFA/MDR space. It appears that the optimal choice of threshold differs slightly between the ARL/TTD space and the NFA/MDR space. There is a clear and expected effect between abrupt and gradual change on the ARL/TTD space mostly in the TTD axis, with TTD being marginally lower for abrupt changes in those detectors whose assumptions are not violated.

Bearing in mind the works of Allipi et al. [17] and Evangelista et al. [16], we were interested in observing the effects of data dimensionality on the missed detection rate. Scatterplots of average missed detection rate against dataset dimensionality, for each category of ensemble and

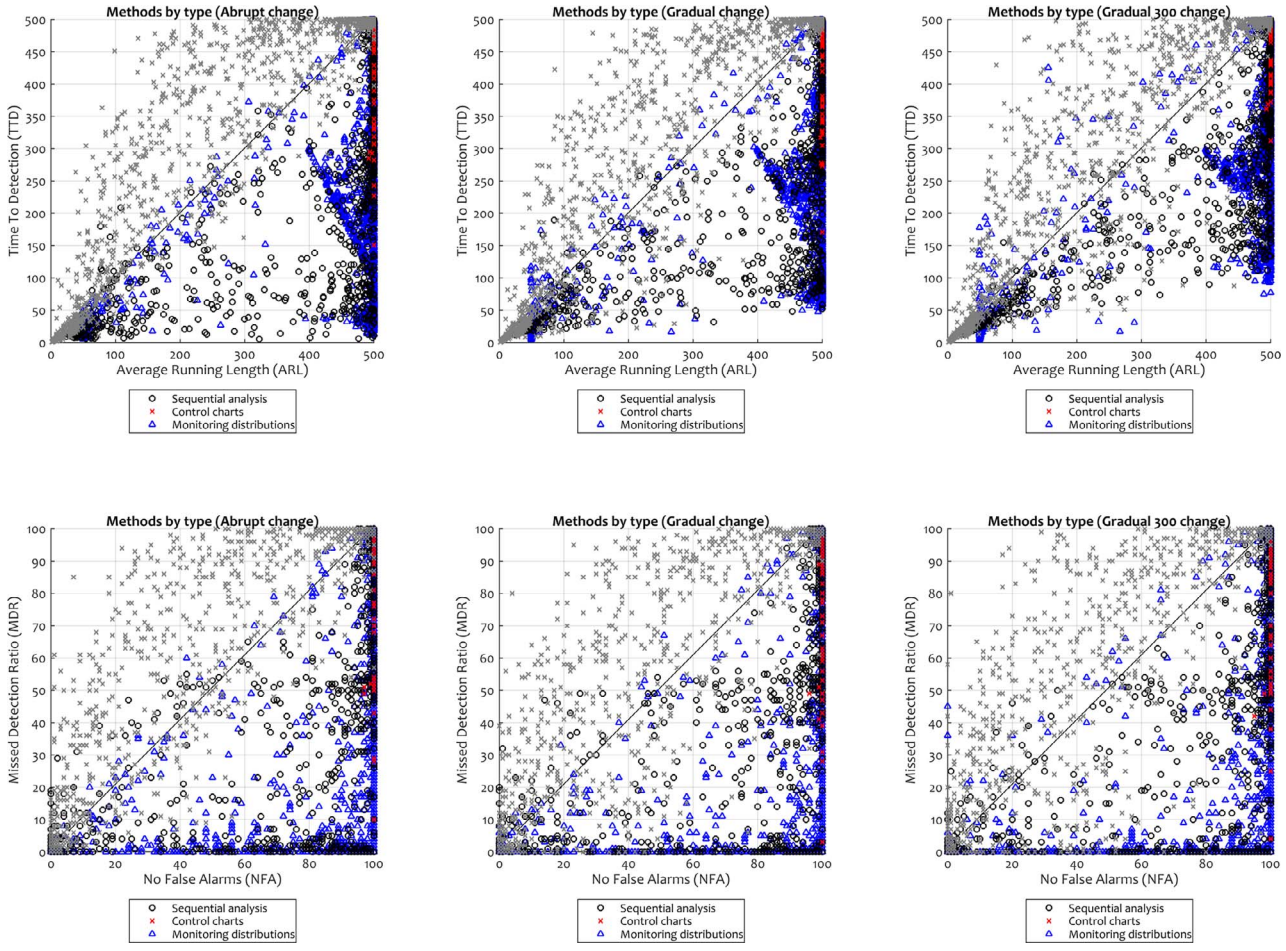

Fig. 3. The three categories of detector, visualised in the ARL/TTD space for the abrupt, gradual 100 and gradual 300 change experiments, respectively. Data points for methods whose assumptions were violated are greyed out, but retain their category marker.

Table 5
The mean and standard deviation of the metrics for each category.

Method	ARL		TTD		NFA		MDR	
	μ	σ	μ	σ	μ	σ	μ	σ
Sequential analysis	433.49	134.28	323.02	187.07	80.21	34.26	59.14	42.26
Control charts	499.93	0.68	486.67	46.14	99.97	0.28	96.46	12.02
Monitoring distributions	435.16	145.36	219.77	176.18	81.07	34.73	29.75	38.82

Table 6
The top 20 performers in the main experiment and the case study. The methods are ranked in the listed 2D spaces by minimum euclidean distance to their respective ideal points, (500, 0), (1, 0), (7684.09, 0) and (0, 0). The ranks of the multivariate detectors and multivariate ensemble are also shown if they were not represented in the top 20.

	Main experiment averages						Case study – KDD Cup 1999					
#	Detector	ARL	TTD	Detector	NFA	MDR	Detector	ARL	TTD	Detector	FPR	MDR
1	SEED-1	484.18	113.07	SEED-5	0.96	0.05	ADWIN-20	10578.05	327.71	HDDMA-1	0.14	0.07
2	SEED-5	494.00	130.66	ADWIN-1	1.00	0.06	SEED-20	10900.19	648.86	CUSUM1-1	0.03	0.26
3	ADWIN-1	499.67	148.46	ADWIN-5	1.00	0.08	SEQ1-5	10930.04	578.64	CUSUM1-5	0.01	0.31
4	CUSUM2-1	462.10	160.48	SEED-1	0.91	0.03	CUSUM1-30	11153.81	1179.54	HDDMA-5	0.03	0.31
5	ADWIN-5	499.91	165.00	SEED-10	0.98	0.14	SEQ1-1	4291.67	180.79	PH-1	0.01	0.32
6	SEED-10	497.54	172.90	ADWIN-10	1.00	0.15	CUSUM2-5	3462.90	1281.90	CUSUM2-1	0.01	0.32
7	PH-1	477.96	187.96	SEQ1-20	0.94	0.18	ADWIN-10	3094.09	85.84	HDDMW-1	0.32	0.08
8	ADWIN-10	499.94	197.93	PH-1	0.86	0.13	SEED-10	2828.08	74.83	GEOMA-1	0.01	0.33
9	SEQ1-5	463.90	242.38	SEQ1-10	0.79	0.06	DDM-5	13724.94	1974.68	MV-50	0.02	0.36
10	SEQ1-10	478.91	247.84	CUSUM2-1	0.75	0.10	HDDMA-10	2605.21	3357.99	Hotelling	0.02	0.36
11	SEQ1-1	453.59	248.14	ADWIN-20	1.00	0.33	CUSUM1-20	2646.27	3734.60	EDDM-1	0.00	0.37
12	CUSUM1-20	374.97	228.50	SEQ1-5	0.64	0.03	ADWIN-5	741.15	48.96	CUSUM1-10	0.00	0.37
13	CUSUM2-5	484.61	264.66	SEQ1-30	0.98	0.37	SEED-5	682.78	48.51	KL	0.01	0.37
14	ADWIN-20	499.99	268.90	CUSUM2-5	0.89	0.37	EDDM-1	563.22	39.67	SPLL	0.02	0.37
15	SEED-20	499.52	274.41	SEED-20	1.00	0.41	DDM-1	441.54	1494.63	EWMA-1	0.00	0.39
16	PH-5	491.43	293.04	PH-5	0.94	0.41	EWMA-1	541.86	2015.76	DDM-1	0.00	0.39
17	SEQ1-20	494.19	294.23	SEQ1-1	0.54	0.03	SEED-1	229.07	32.73	ADWIN-1	0.01	0.40
18	CUSUM1-10	219.09	114.13	ADWIN-30	1.00	0.50	ADWIN-1	187.46	24.95	SEED-1	0.00	0.41
19	CUSUM1-30	439.02	308.69	CUSUM1-20	0.59	0.34	PH-1	113.59	15.71	SEED-5	0.00	0.43
20	ADWIN-30	499.99	328.74	CUSUM1-30	0.80	0.52	GEOMA-1	108.70	19.54	ADWIN-5	0.00	0.44
		⋮	⋮		⋮	⋮		⋮	⋮			
	# Detector	ARL	TTD	# Detector	NFA	MDR	# Detector	ARL	TTD			
	21 Hotelling	499.95	432.97	30 Hotelling	0.01	0.01	23 KL	∞	∞			
	34 SPLL	484.61	264.66	47 SPLL	0.04	0.04	25 SPLL	∞	∞			
	39 MV-50	499.88	497.29	54 MV-50	1.00	1.00	26 MV-50	541.86	2015.76			
	47 KL	57.02	56.73	68 KL	0.86	0.13	27 Hotelling	9137.79	8020.57			

for the multivariate detectors, are presented in Fig. 6. The scatter patterns suggest that changes in higher-dimensional spaces are more likely to be missed.

4.1. The case study

The right half of Table 6 summarises the top 20 performers on the case study data. As this experiment was a single run, we present the false positive rate as FPR, instead of the NFA measure. The methods were ranked by the minimum euclidean distance to the ideal points (7864.09, 0) and (0, 0) for the ARL/TTD and FPR/MDR spaces respectively. The ideal ARL of 7864.09 was calculated by observing the ARL of a perfect, 'cheating' detector, which signalled immediately for all changepoints and recorded no false positives. We see a familiar pattern in the ARL/TTD space, with the SEED, ADWIN and CUSUM-based methods well represented within the top 20. In the FPR/MDR space, the winners are primarily low-threshold ensembles. We note that 8 methods; ADWIN-1, ADWIN-5, SEED-1, SEED-5, EDDM-1, pH-1, GEOMA-1 and EWMA-1 are represented in the top 20 in both spaces. We also observe that the top ranked ensembles across the two spaces here differed modestly from the top performers in the main experiment with the simulated abrupt and gradual changes. The improvement in performance of control chart-based methods may be due to the incidence of a number of contextually important binary features in this dataset. The best performing multivariate detectors were ranked 23rd and 9th in the two spaces respectively. Apart from the high false positive rates of $HDDM_W$–1 and $HDDM_A$–1, the ensembles were competitive or better than the multivariate detectors on TTD and MDR, and generally exhibited less false positives. The dominance of the low-threshold ensembles mirrors their success in the previous experiment, and suggests that between 1% and 5% agreement is a sensible starting point when employing this scheme, across a range of different detectors.

5. Conclusions

The results of the experiment and the case study demonstrate the viability of ensemble combination of univariate change detectors over multivariate data. Over 96 datasets, ensemble methods frequently outperformed multivariate detectors in all metrics, especially at low agreement thresholds. The multivariate detectors did not even feature in the top 20 overall performers in either space, as seen in Table 6. This would appear to tally with the conclusions of Evangelista et al. [16]. The SEED and ADWIN detectors appear to be the best suited to ensemble combination in this manner. Given that the SEQ1 algorithm is an ADWIN-derivative, we would expect it to exhibit a similar performance. We see that it does exhibit very similar performance to the ADWIN ensembles in terms of missed detections, but it signals far more eagerly for a higher rate of false alarms. This may be a reflection, as we noted in Section 3.1.3, of the authors' choice of the Bernstein bound over the more conservative Hoeffding bound to set the threshold.

Those detectors which make strong assumptions on the basis that they are monitoring the error stream of an attached learner were unsurprisingly poor when applied to raw data in this scheme. This accounts for the worse-than-chance performances of the $HDDM_A$, $HDDM_W$, EDDM, DDM and EWMA methods.

Upon observation of the results, we note that the ideal agreement threshold varies between detectors. The curves in Figs. 4 and 5 can be used to pick a suitable threshold for each of the successful detectors. Taking ADWIN for example, the lack of movement on the false alarm rate relative to the threshold changes suggests that an ensemble might be close to optimal if any member is given absolute authority for signalling. As a counter example, the SEQ1 ensembles seem to have an optimal agreement threshold of between 10% and 20%.

We observed empirically that all categories of detectors exhibited a positive relationship between missed detections and dataset dimensionality, as Allipi et al. [17] suggest, albeit to varying degrees. Evangelista et al. [16] also state that unsupervised learning in subspaces of the data is a means to address the curse of dimensionality. This is not strongly reflected in Fig. 6, with the multivariate detectors

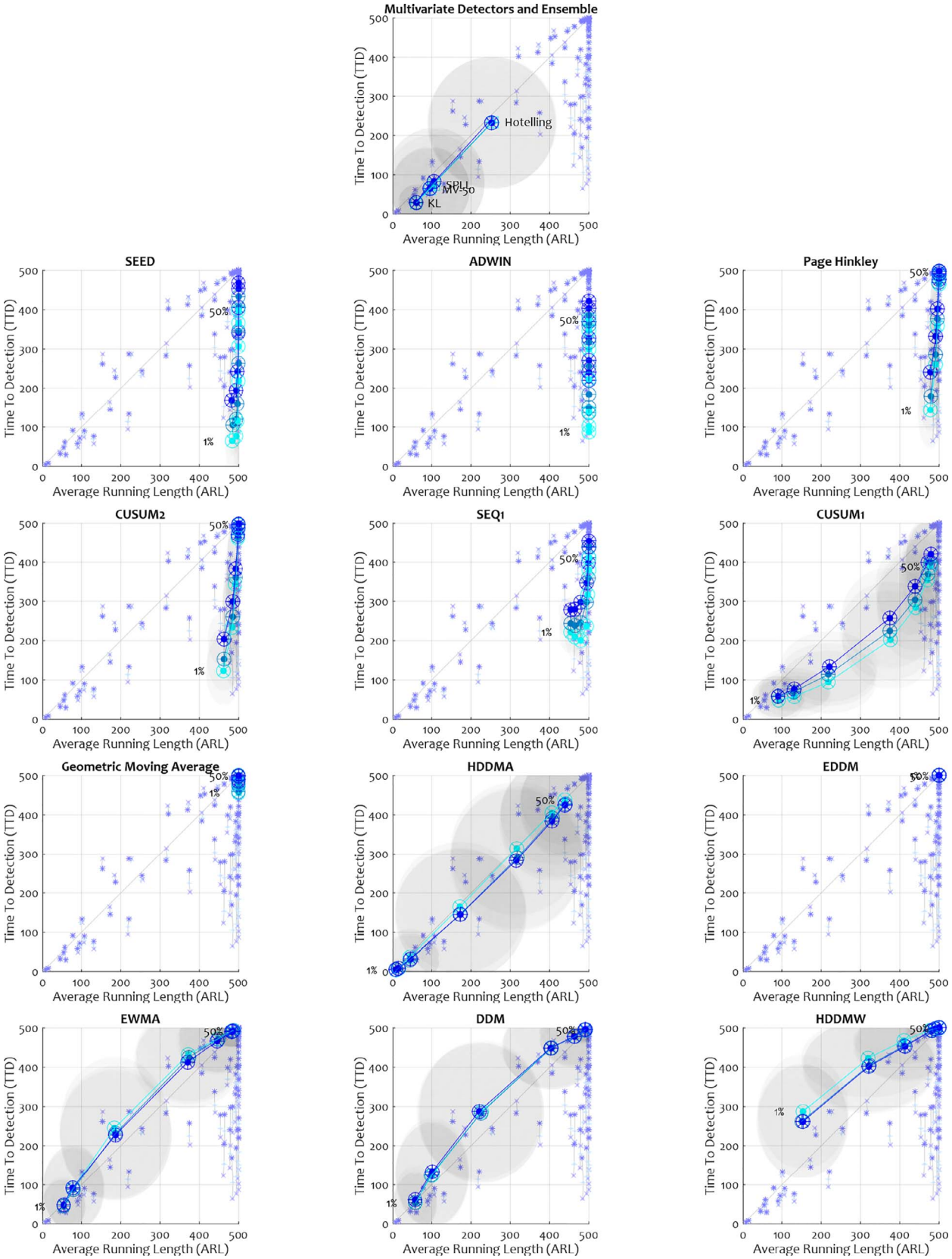

Fig. 4. Change detection methods in the space spanned by ARL and TTD for the main experiment. Each method has been examined with different agreement thresholds. Each plot contains 88 gradual and 88 abrupt detector points, averaged across the 96 data sets – gradual 300 as a blue × (darkest), linked to the paired gradual 100 result as a purple + and the abrupt result as a cyan ⋆ (lightest). Each detector's points are highlighted, again in blue, purple and cyan for gradual 300, gradual and abrupt change type, respectively. The shaded ellipses around the mean detector results are the standard deviations across the 96 datasets. The ideal point is (500, 0). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

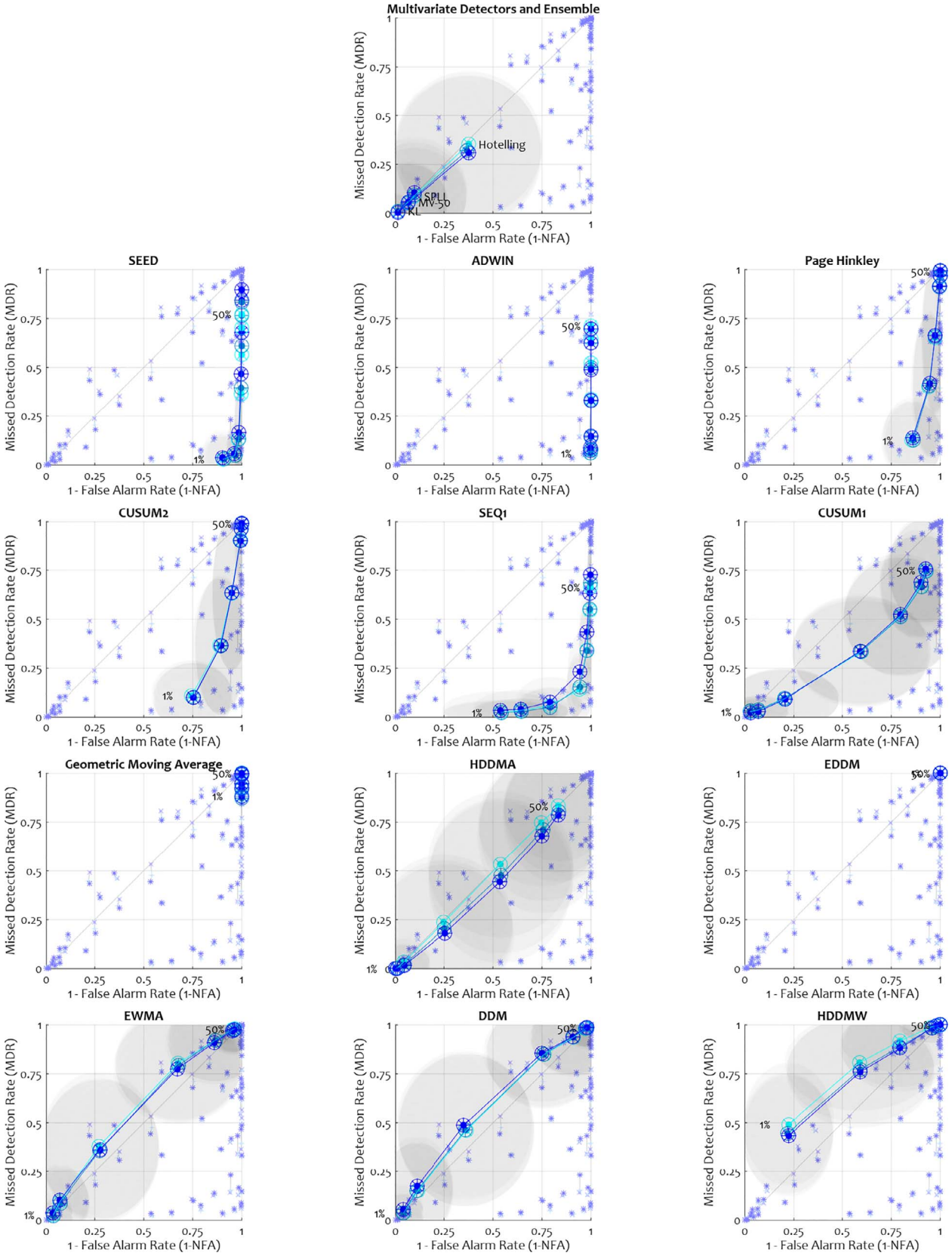

Fig. 5. Change detection methods in the space spanned by NFA and MDR for the main experiment. Each method has been examined with different agreement thresholds. Each plot contains 88 gradual and 88 abrupt detector points, averaged across the 96 data sets – gradual 300 as a blue × (darkest), linked to the paired gradual 100 result as a purple + and the abrupt result as a cyan ⋆ (lightest). Each detector's points are highlighted, again in blue, purple and cyan for gradual 300, gradual 100 and abrupt change type, respectively. The shaded ellipses around the mean detector results are the standard deviations across the 96 datasets. The ideal point is (1, 0). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

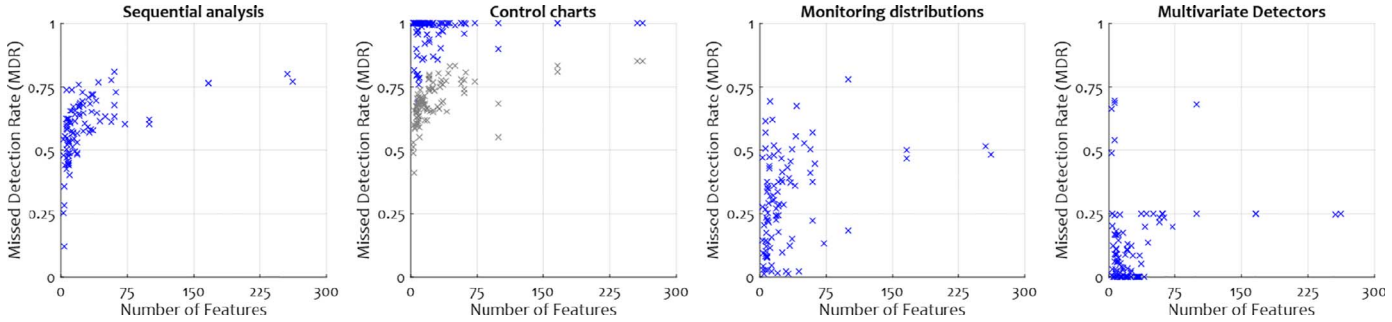

Fig. 6. Scatter plots of dataset dimensionality against average missed detection rate for the 96 datasets. The plots are arranged by the category of the detectors. Data points from detectors with violated assumptions are greyed out.

appearing to exhibit the weakest relationship of missed detections with dimensionality. However, the ensembles had a much wider spread of results, and the better ensembles considerably outperformed the multivariate detectors.

The experimental results invite many avenues of future work. The application of existing work on feature extraction, weighting or selection could change the optimal ensemble thresholds by removing redundant features. Ensembles could be tailored to the type and rate of the expected changes in the data stream, incorporating domain-specific knowledge rather than the generic approach here. The numerous univariate change detection approaches not considered within this paper can be evaluated in similarly constructed ensembles.

Acknowledgments

This work was done under project RPG-2015-188 funded by The Leverhulme Trust, UK; Spanish Ministry of Economy and Competitiveness through project TIN 2015-67534-P and the Spanish Ministry of Education, Culture and Sport through Mobility Grant PRX16/00495. The 96 datasets were originally curated for use in the work of Fernández-Delgado et al. [53] and accessed from the personal web page of the author[5]. The KDD Cup 1999 dataset used in the case study was accessed from the UCI Machine Learning Repository [10].

References

[1] E. Page, Continuous inspection schemes, Biometrika 41 (1) (1954) 100–115, http://dx.doi.org/10.2307/2333009.
[2] A. Wald, Sequential Analysis, Dover Publications, New York, 1974.
[3] M. Basseville, I.V. Nikiforov, Detection of Abrupt Changes: Theory and Application, vol. 104, Prentice Hall, Englewood Cliffs, 1993.
[4] M. Markou, S. Singh, Novelty detection: a review - Part 1: statistical approaches, Sig. Process. 83 (12) (2003) 2481–2497.
[5] M.A.F. Pimentel, D.A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, Sig. Process. 99 (2014) 215–249, http://dx.doi.org/10.1016/j.sigpro.2013.12.026.
[6] I. Ben-gal, Outlier detection, Data Min. Knowl. Discov. Handb. (2005) 131–146, http://dx.doi.org/10.1007/0-387-25465-x_7.
[7] A. Bifet, G. Holmes, R. Kirkby, B. Pfahringer, MOA massive online analysis, J. Mach. Learn. Res. 11 (2011) 1601–1604. [Online]. Available: http://moa.cs.waikato.ac.nz/details/ .
[8] A. Bifet, J. Read, B. Pfahringer, G. Holmes, I. Žliobaitė, CD-MOA: change detection framework for massive online analysis, Adv. Intell. Data Anal. XII (2013) 92–103, http://dx.doi.org/10.1007/978-3-642-41398-8_9.
[9] L. Kuncheva, Classifier ensembles for detecting concept change in streaming data: overview and perspectives, Proceedings of the Second Workshop SUEMA, ECAI 2008, (2008), pp. 5–9. [Online]. Available: http://hdl.handle.net/10242/41809 .
[10] M. Lichman, {UCI} Machine Learning Repository, 2013. [Online]. Available: https://archive.ics.uci.edu/ml/citation_policy.html.
[11] L.I. Kuncheva, Change detection in streaming multivariate data using likelihood detectors, IEEE Trans. Knowl. Data Eng. 25 (5) (2013) 1175–1180, http://dx.doi.org/10.1109/tkde.2011.226.
[12] F. Zorriassatine, A. Al-Habaibeh, R.M. Parkin, M.R. Jackson, J. Coy, Novelty detection for practical pattern recognition in condition monitoring of multivariate processes: a case study, Int. J. Adv. Manuf. Technol. 25 (9–10) (2005) 954–963.
[13] D. Agarwal, An Empirical Bayes approach to detect anomalies in dynamic multidimensional arrays, Proceedings - IEEE International Conference on Data Mining, ICDM, (2005), pp. 26–33.
[14] T.D. Nguyen, M.C. Du Plessis, T. Kanamori, M. Sugiyama, Constrained least-squares density-difference estimation, IEICE Trans. Inf. Syst. 97 (2014) 1822—1829.
[15] A.G. Tartakovsky, B.L. Rozovskii, R.B. Blažek, H. Kim, Detection of intrusions in information systems by sequential change-point methods, Stat. Methodol. 3 (3) (2006) 252–293.
[16] P.F. Evangelista, M.J. Embrechts, B.K. Szymanski, Taming the curse of dimensionality in kernels and novelty detection, Applied Soft Computing Technologies: The Challenge of Complexity, Springer, 2006, pp. 425–438.
[17] C. Alippi, G. Boracchi, D. Carrera, M. Roveri, Change detection in multivariate datastreams: likelihood and detectability loss, Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, AAAI Press, 2016, pp. 1368–1374.
[18] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (4) (2014) 1–37, http://dx.doi.org/10.1145/2523813.
[19] G. Ditzler, M. Roveri, C. Alippi, R. Polikar, Learning in nonstationary environments: a survey, IEEE Comput. Intell. Mag. 10 (4) (2015) 12–25.
[20] L. Tarassenko, A. Hann, D. Young, Integrated monitoring and analysis for early warning of patient deterioration. Br. J. Anaesth. 97 (1) (2006) 64–68, http://dx.doi.org/10.1093/bja/ael113.
[21] X. Song, M. Wu, C. Jermaine, S. Ranka, Statistical change detection for multi-dimensional data, Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '07, V (2007), p. 667. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1281192.1281264 .
[22] T. Dasu, S. Krishnan, S. Venkatasubramanian, K. Yi, An information-theoretic approach to detecting changes in multi-dimensional data streams, Proc. Symp. on the Interface of Statistics, Computing Science, and Applications, (2006).
[23] G. Krempl, Z.F. Siddiqui, M. Spiliopoulou, Online clustering of high-dimensional trajectories under concept drift, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 261–276.
[24] M.M. Gaber, P.S. Yu, Classification of changes in evolving data streams using online clustering result deviation, Proc. Of International Workshop on Knowledge Discovery in Data Streams, (2006).
[25] H. Hotelling, The generalization of Student's ratio, Breakthroughs in Statistics, (1992), p. 54—-65.
[26] W.J. Faithfull, L.I. Kuncheva, On optimum thresholding of multivariate change detectors, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8621 LNCS, Springer Verlag, 2014, pp. 364–373.
[27] B. Krawczyk, L.L. Minku, J. Gama, J. Stefanowski, M. Woźniak, Ensemble learning for data stream analysis: a survey, Inf. Fus. 37 (2017) 132–156.
[28] H.M. Gomes, J.P. Barddal, F. Enembreck, A. Bifet, A survey on ensemble learning for data stream classification, ACM Comput. Surv. 50 (2) (2017) 23.
[29] B.I.F. Maciel, S.G.T.C. Santos, R.S.M. Barros, A lightweight concept drift detection ensemble, 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2015, pp. 1061–1068, http://dx.doi.org/10.1109/ICTAI.2015.151.
[30] M. Woźniak, P. Ksieniewicz, B. Cyganek, K. Walkowiak, Ensembles of heterogeneous concept drift detectors-experimental study, IFIP International Conference on Computer Information Systems and Industrial Management, Lecture Notes in Computer Science vol. 9842, Springer, 2016, pp. 538–549.
[31] L. Du, Q. Song, L. Zhu, X. Zhu, A selective detector ensemble for concept drift detection, Comput. J. 58 (3) (2014) 457–471.
[32] C. Alippi, G. Boracchi, M. Roveri, Hierarchical change-detection tests, IEEE Trans. Neural Netw. Learn. Syst. 28 (2) (2017) 246–258.
[33] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, R. Gavaldà, New ensemble methods for evolving data streams, Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2009, pp. 139–148.
[34] A. Bifet, E. Frank, G. Holmes, B. Pfahringer, Accurate ensembles for data streams: combining restricted hoeffding trees using stacking. 2nd Asian Conference on Machine Learning (ACML2010), (2010), pp. 1–16.
[35] A. Bifet, E. Frank, G. Holmes, B. Pfahringer, Ensembles of restricted hoeffding trees, Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, 15 (2012), pp. 434–442.

[5] http://persoal.citius.usc.es/manuel.fernandez.delgado/papers/jmlr/

[36] I. Frias-Blanco, J. del Campo-Avila, G. Ramos-Jimenez, R. Morales-Bueno, A. Ortiz-Diaz, Y. Caballero-Mota, Online and non-parametric drift detection methods based on Hoeffding's bounds, IEEE Trans. Knowl. Data Eng. 27 (3) (2015) 810–823, http://dx.doi.org/10.1109/tkde.2014.2345382.
[37] J. Gama, P. Medas, G. Castillo, P. Rodrigues, Learning with drift detection, Adv. Artif. Intell. - SBIA 2004 (2004) 286–295, http://dx.doi.org/10.1007/978-3-540-28645-5_29.
[38] M. Baena-García, J. del Campo Ávila, R. Fidalgo, A. Bifet, R. Gavalda, R. Morales-Bueno, Early drift detection method, Fourth International Workshop on Knowledge Discovery from Data Streams, 6 (2006), pp. 77–86.
[39] W.N. Street, Y. Kim, A streaming ensemble algorithm (SEA) for large-scale classification, Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '01, 4 (2001), pp. 377–382. [Online]. Available: http://portal.acm.org/citation.cfm?doid=502512.502568 .
[40] D.T.J. Huang, Y.S. Koh, G. Dobbie, R. Pears, Detecting volatility shift in data streams, Data Mining (ICDM), 2014 IEEE International Conference on, IEEE, 2014, pp. 863–868.
[41] A. Bifet, R. Gavaldà, Learning from time-changing data with adaptive windowing, Proceedings of the 2007 SIAM International Conference on Data Mining, (2007), pp. 443–448, http://dx.doi.org/10.1137/1.9781611972771.42.
[42] S. Sakthithasan, R. Pears, Y.S. Koh, One pass concept change detection for data streams, Adv. Knowl. Discov. Data Min. (2013) 461–472, http://dx.doi.org/10.1007/978-3-642-37456-2_39.
[43] G.J. Ross, N.M. Adams, D.K. Tasoulis, D.J. Hand, Exponentially weighted moving average charts for detecting concept drift, Pattern Recognit. Lett. 33 (2) (2012) 191–198, http://dx.doi.org/10.1016/j.patrec.2011.08.019.
[44] S.W. Roberts, Control chart tests based on geometric moving averages, Technometrics 3 (2012) 239–250, http://dx.doi.org/10.2307/1271439.
[45] H. Mouss, D. Mouss, N. Mouss, L. Sefouhi, Test of page-hinckley, an approach for fault detection in an agro-alimentary production system, Proceedings of the 5th Asian Control Conference, (2004), pp. 815–818.
[46] J. Gama, R. Sebastião, P.P. Rodrigues, On evaluating stream learning algorithms, Mach. Learn. 90 (3) (2013) 317–346.
[47] R. Klinkenberg, T. Joachims, Detecting concept drift with support vector machines, Proceedings of ICML-00, 17th International Conference on Machine Learning, (2000), pp. 487–494.
[48] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, Mach. Learn. 23 (1) (1996) 69–101.
[49] L. Kuncheva, That elusive diversity in classifier ensembles, Iberian Conference on Pattern Recognition and Image, vol. 2652, Springer, 2003, pp. 1126–1138. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-44871-6_130 .
[50] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms: Second Edition, Wiley Blackwell, 2014.
[51] D. Brzezinski, J. Stefanowski, Ensemble diversity in evolving data streams, International Conference on Discovery Science, Lecture Notes in Computer Science vol. 9956, Springer, 2016, pp. 229–244.
[52] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the kdd cup 99 data set, Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on, IEEE, 2009, pp. 1–6.
[53] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems, J. Mach. Learn. Res. 15 (1) (2014) 3133–3181.