Hierarchical reduced-space drift detection framework for multivariate supervised data streams

Zhang, Shuyi; Tino, Peter; Yao, Xin

DOI:
10.1109/TKDE.2021.3111756

Hierarchical Reduced-Space Drift Detection Framework for Multivariate Supervised Data Streams

Shuyi Zhang, Peter Tino, and Xin Yao, *Fellow, IEEE*

Abstract—In a streaming environment, the characteristics of the data themselves and their relationship with the labels may change over time. Most drift detection methods for supervised data streams are performance-based, that is, they detect changes only after the classification accuracy deteriorates. This may not be sufficient in many application areas where the reason behind a drift is also important. Another category of drift detectors are data distribution-based detectors. Although they can detect some drifts within the input space, changes affecting only the labelling mechanism cannot be identified. Furthermore, little work is available on drift detection for high-dimensional data streams. In this paper we propose an advanced **H**ierarchical **R**educed-space **D**rift **D**etection (HRDD) framework for supervised data streams which captures drifts regardless of their effects on classification performance. This framework suggests monitoring both marginal and class-conditional distributions within a lower-dimensional space specifically relevant to the assigned classification task. Experimental comparisons have demonstrated that HRDD not only achieves high-quality performance on high-dimensional data streams, but also outperforms its competitors in terms of detection recall, precision and F-measure across a wide range of different concept drift types including subtle drifts.

Index Terms—Concept drift, drift detection, data stream mining, online learning

1 Introduction

In real-world applications such as weather prediction, industrial quality control and fraud detection, data often arrives in the form of a stream. Data streams are likely to be time-varying. *Concept drift* refers to a change in the underlying data distribution and/or its relationship with the target label [1]. This problem has received growing attention not only because it may greatly harm the reliability of real time machine learning systems, but also because it is of practical importance to understand the nature and the reason of the change [2]. One way of categorizing drift is by its influence on the target concept. Changes affecting the posterior class probabilities $P(Y|\mathbf{X})$ are called real drifts, whereas changes affecting the input distribution $P(\mathbf{X})$ only are called virtual drifts [3].

Various detection methods have been proposed to explicitly mark out the drifts [4], [5]. Nonetheless, current methods cannot well address both types of drifts simultaneously. They monitor over time either some classification performance-related indicators [6], [7], [8], [9], [10], or some data distribution-related characteristics [11], [12], [13], [14], [15], [16]. Existing detectors for supervised data streams primarily belong to the former category [4], [17]. They concentrate on addressing real drifts which lead to a decline in classification performance only. Two popular algorithms within this category are drift detection method (DDM) [6] and early drift detection method (EDDM) [7]. DDM detects abrupt drift by applying statistical test on the false classification rate directly, whereas EDDM monitors the distance between consecutive classification errors. Linear four rate (LFR) [10] is another detector which monitors all components of the confusion matrix. Although these detectors can be used in conjunction with any classifier since they utilize only the error stream, their detection performance is still dependent on the chosen base classifier [18]. Besides, they neglect drifts not deteriorating the classification performance, which may harm the interpretation of the data.

Unlike the above, change detection tests (CDTs) within the latter category choose to monitor some underlying data features regardless of label information. For instance, cumulative sum (CUSUM) control chart [11] keeps track of the cumulative sum of deviations, and intersection of confidence intervals (ICI) CDT [13] carefully designs mean and variance-related features that follow a Gaussian distribution. For multivariate data streams, detectors either compare the estimated empirical density of two windows [12], [16] or conduct univariate CDTs for each individual dimension [13], [19]. These approaches tend to be problematic for

- *Shuyi Zhang and Xin Yao are with the Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology (SUSTech), Shenzhen, Guangdong 518055, China, and also with the CERCIA, School of Computer Science, University of Birmingham, B15 2TT Birmingham, U.K. E-mail: sxz745@bham.ac.uk, xiny@sustech.edu.cn.*
- *Peter Tino is with the CERCIA, School of Computer Science, University of Birmingham, B15 2TT Birmingham, U.K. E-mail: P.Tino@cs.bham.ac.uk.*

Manuscript received 4 Aug. 2020; revised 19 June 2021; accepted 24 Aug. 2021. Date of publication 16 Sept. 2021; date of current version 3 Feb. 2023.
(Corresponding author: Xin Yao.)
Recommended for acceptance by J. Lee.
Digital Object Identifier no. 10.1109/TKDE.2021.3111756

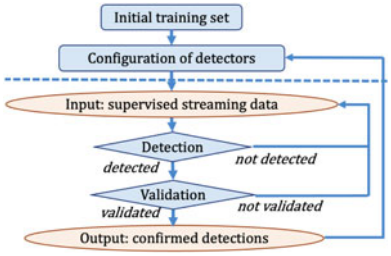

Fig. 1. General framework of HCDT [21].

higher-dimensional data streams [9], [10]. Besides, while these detectors can be directly applied to supervised data streams, they do not consider any class information and thus cannot detect real drifts affecting the data labelling mechanisms only (e.g., a class swap) [20]. In addition, most methods monitor the overall input space, hence they tend to be insensitive to drifts affecting a sub-region only (e.g., a single class drift).

Based on the vast scope of individual detectors existing in the literature, more consolidated frameworks have been developed recently. Hierarchical change detection test (HCDT) presented in Fig. 1 is a general two-layered detect-and-verify framework [21]. HCDT incorporates in Layer-I a simple non-parametric online detector such as the CUSUM or ICI CDT, and in Layer-II an offline two-sample test such as the Hotelling T2 test [22]. Once a potential drift is reported in Layer-I, Layer-II is activated to compare the training set with the most recent set so as to confirm (or deny) the validity of the suspected drift. HCDT has been shown to achieve more advantageous false positive rate (FPR) versus detection delay (DD) trade-off than its single CDT counterpart, but it has only been tested on non-labelled scalar data [21]. Direct application of this framework to multivariate supervised data streams still suffers from the aforementioned deficiencies of distribution-based detectors.

Inspired by this framework, another hierarchical framework named HLFR for supervised data streams is proposed [23]. HLFR incorporates LFR as the base detector in Layer-I and a permutation test in Layer-II. However, HLFR is purely classification performance-based, therefore it cannot detect real and virtual drifts simultaneously. Different from the above-mentioned hierarchical frameworks, another multiple-layered drift detection algorithm is also proposed [24]. This test individually addresses the label drift, feature drift and the decision boundary in three sequential layers. The decision criteria are based on Information Value and Jaccard similarity (IV-Jac). However, this algorithm tackles the challenge of sparseness and high dimensionality of text data streams. It is more suitable for data with discrete or categorical features.

Concept drifts can be incurred by many causes, and they may present differently in different time periods [6]. Therefore, it is important to be aware of all drifts regardless of their effects on classification, especially in areas such as condition monitoring, adversarial attack detection and strategic planning. Furthermore, there is little work on drift detection on high-dimensional continuous data streams. In this paper, we adopt the hierarchical structure and propose a new detection framework, HRDD (**H**ierarchical **R**educed-space **D**rift **D**etection framework), to detect both real and virtual drift accurately and efficiently for multi-dimensional data streams. The key idea is to leverage the knowledge from supervised information to discover changes that may not be detected by the existing detection methods. To achieve this goal, first, a lower-dimensional feature space for the given classification task is explicitly constructed using the stationary training data. Each incoming data is projected to this space upon arrival. Next, we monitor not only the marginal distribution of the data stream, but also each individual class-conditional distribution. Finally, a novel method to reconfigure more informative retraining datasets after each detection is presented. HRDD can be used in conjunction with any base CDT and classifier, and the performance is independent of the choice of the classifier. The contributions of our work include:

1) A new hierarchical detection framework proposed for supervised data streams that detects both real and virtual drifts.
2) Compared with the existing HCDT framework, HRDD is more accurate and efficient in terms of a high number of true detections, while maintaining a low number of false alarms, when operating on higher-dimensional data streams.
3) For both real and virtual drifts, HRDD performs no worse, and in many cases better, than state-of-the-art detection algorithms, whether they are performance-based or distribution-based, in terms of more true detections and fewer false alarms within any specified acceptable detection delay range.

It is worth pointing out that detecting concept drifts and adapting the classification model to the data are two different mechanisms. From the practical point of view, an accurate detector is crucial for maintaining good classification performance in the long run. The focus of this paper is to detect drifts. How to build an appropriate classifier for a specific data stream is beyond the scope of this study.

The rest of the paper is organized as follows. *Section 2* formulates the problem of concept drift. *Section 3* explains each component of the proposed HRDD framework in detail. In *Section 4*, four sets of experiments are carried out on both synthetic and real-world data streams to demonstrate the superiority of HRDD in comparison with some state-of-the-art detectors, including both data distribution-based and classification performance-based ones. *Section 5* concludes the paper and points out potential future extensions of this work.

2 Terminology and Problem Formulation

In a streaming environment, a supervised data stream to be inspected for change is formed by observations $\{(\mathbf{x}_t, y_t), t \in \mathbb{Z}^+\}$. $\mathbf{x}_t \in \mathbb{R}^d$ represents the d-dimensional feature vector of the observation at time t and y_t is its class label. $y_t \in \{0, 1, \ldots, Q\}$ where $Q+1$ is the number of available classes. For a binary classification task, $y_t \in \{0, 1\}$. The generation process of the observations can be denoted by the joint distribution $P(\mathbf{X}, Y)$. A concept drift is said to occur when there is a change in the joint probability $P(\mathbf{X}, Y)$ [17].

More specifically, consider a single drift scenario, concept drift detection aims to find out the unknown change point T^* where $(\mathbf{x}_t, y_t) \sim P_{\mathcal{U}}(\mathbf{X}, Y)$ for $t < T^*$, $(\mathbf{x}_t, y_t) \sim P_{\mathcal{V}}(\mathbf{X}, Y)$ for $t \geq T^*$ and $P_{\mathcal{V}}(\mathbf{X}, Y) \neq P_{\mathcal{U}}(\mathbf{X}, Y)$. Analogous description can be made for multiple drift scenarios.

The joint probability $P(\mathbf{X}, Y)$ can be written as

$$P(\mathbf{X}, Y) = P(Y|\mathbf{X}) \cdot P(\mathbf{X}), \tag{1}$$

where $P(\mathbf{X})$ can be obtained through marginalization

$$P(\mathbf{X}) = \sum_{q=0}^{Q} P(Y = q) \cdot P(\mathbf{X}|Y = q). \tag{2}$$

Based on the probabilistic definition of a concept drift and the above decomposition, it is not difficult to tell that the change can manifest itself in different forms corresponding to the different components of the joint probability [25], [26]. Assuming $P(Y)$ is stationary over time, drift can occur in: 1) the marginal distribution over covariates $P(\mathbf{X})$; 2) the posterior class probability or classification concept $P(Y|\mathbf{X})$; 3) one or more class-conditional distributions $P(\mathbf{X}|Y)$.

Most existing work tackling drift in supervised data streams focus on the second type of drifts (or real drifts), since it is considered to be the most detrimental to classification accuracy. However, we consider the detection of all types of drift to be equally important for the following reasons. First, even when a so-called virtual drift takes place and classification accuracy is not negatively affected, the optimal decision boundary is often likely to change. Retraining the classifier can still improve classification performance. Second, detection of such drifts provides insight into the underlying data streams, which can help understanding the behavior of the data generation source. This information may also be beneficial when there is a pattern in a series of multiple drifts. The systematic study [4] supported the view that all types of change are equally important, but also claimed that there is a lack of research effort in the investigation of drifts not affecting classification accuracy (or virtual drifts).

Therefore, in this paper we do not explicitly distinguish between real and virtual drifts. We present a framework aiming to detect all types of drifts regardless of whether they affect classification or not. Then, practitioners can decide whether it is worth modifying the current classification model based on the specific application scenario.

3 Hierarchical Reduced-Space Drift Detection Framework for Multivariate Supervised Data Streams

In this section we describe a novel drift detection framework named HRDD (**H**ierarchical **R**educed-space **D**rift **D**etection framework) aiming to answer the following research questions.

1) How to detect both real and virtual drifts in supervised data streams regardless of their effect on classification performance?
2) How to improve the efficiency of data distribution-based detector for high-dimensional data streams?
3) How to improve detection performance to achieve high true detections and low false alarms within a specified delay range for all types of drifts even when the magnitude of drift is small?

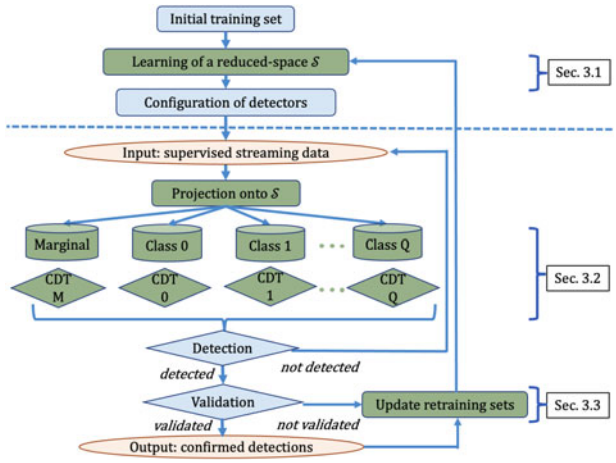

Fig. 2. General framework of our proposed HRDD. Detailed descriptions for each novel component are provided in Sections 3.1, 3.2 and 3.3.

HRDD adopts the hierarchical structure introduced in [21] but with three major novel components, which will be explained in this section. The general outline of HRDD is presented in Fig. 2. The algorithmic version of HRDD is presented in Algorithm 1. This framework has a high degree of flexibility and may be customized effortlessly. Since there are no assumptions on the multivariate data streams, any detection and validation tests can be used as long as they are capable of detecting the same type of change. The choice of individual test is independent of our proposed strategies. Although we provide one possible realization for a binary classification problem as an illustrative example in this paper, it is worth noting that the general framework of HRDD is also suitable for multi-class data streams.

Algorithm 1. General Framework of HRDD

Input: initial training sets TS^M for the marginal CDT and $TS^0, TS^1, \ldots$ for the class-conditional CDTs
Output: confirmed detections

1 Find the lower-dimensional feature space $\mathcal{S}$;
2 Initialize the marginal and class-conditional CDTs with TS^M and $TS^0, TS^1, \ldots$ respectively;
3 **while** *there is incoming data* **do**
4 Project data onto $\mathcal{S}$;
5 Perform concept drift detection within $\mathcal{S}$;
6 **if** *a change is detected by any of the CDTs at* $\widehat{T}$ **then**
7 Estimate the potential drift starting point T_{ref};
8 Activate the validation layer on the respective stream;
9 **if** *change is validated* **then**
10 Record $\widehat{T}$ as a confirmed detection;
11 Define TS_C^M as $\{\mathbf{x}_t | t \in [T_{ref}, \ldots, \widehat{T}]\}$;
12 Update training sets $TS^M, TS^0, TS^1, \ldots$ accordingly and continue from line 1.
13 Output the confirmed changes.

3.1 Learning of a Lower-Dimensional Subspace

In this module, we take the information from class labels into consideration and propose a preprocessing step specifically designed for drift detection for supervised data streams. The

aim of this step is to identify a lower-dimensional feature space $\mathcal{S}$ that contains the most relevant information for the given classification task. By identifying such a subspace spanned by the training samples (line 1, Algorithm 1), incoming multivariate data samples can be easily projected onto this space (line 4, Algorithm 1). Then, instead of monitoring the original input space, the detection is carried out within this reduced feature space for the particular classification task. Comparing with the existing HCDT without this step, HRDD inherently reduces the possibility of false alarms as well as the computational burden because there are fewer dimensions to examine. Meanwhile, valuable data characteristics relevant to classification are preserved.

It is worth noting that subspace selection methods have been used for change detection in signal processing applications [29], [30]. However, how a change is defined in such applications is very different than that in our setting. Consequently, the characteristics that the subspace shall possess also vary. HRDD combines the information from both original data space and the label space to identify the most appropriate subspace for concept drift detection. Besides, many subspace-based change detection algorithms for time series data make particular assumptions on their data streams [31], [32]. For instance, in [32], the data stream is assumed to follow a Gaussian distribution. HRDD does not make any assumptions on either the data stream or the underlying subspace.

As one possible realization of HRDD within a bi-class setting, we choose a recursive support vector machine (RSVM) [27] as a tool for identification of the relevant reduced-space $\mathcal{S}$. The detailed RSVM algorithm is presented in Algorithm 2, where l is the length of an initial training dataset and $\phi(\cdot)$ is the kernel function. RSVM was initially proposed for both dimensionality reduction and accuracy improvement for offline classification problems. It starts as a regular SVM [28] but can recursively derive new maximum margin features. The dimension R of the reduced-space can either be set by the practitioner a priori, or be automatically identified when the number of components is sufficient to account for most of the differences in the classification task.

Algorithm 2. RSVM [27]

input: training set TS^M of length l ; the desired dimension of the reduced-space R (or threshold ϵ)

output: projectors $\{\mathbf{w}_r \in \mathbb{R}^d | \, r = 1, \ldots, R\}$

1 Determine the vector $\tilde{\mathbf{w}}_1 = \sum_{i=1}^{l} \alpha_i^1 \phi(\mathbf{x}_i)$ by solving the dual optimization problem [28];

2 Let $\mathbf{w}_{r-1} = \tilde{\mathbf{w}}_{r-1}/||\tilde{\mathbf{w}}_{r-1}||$ and generate the following training set for SVM problem by projecting the training samples onto a subspace that is orthogonal to

$$\mathbf{w}_{r-1} : \phi(\mathbf{x}_i^r) = \phi(\mathbf{x}_i^{r-1}) - \langle \phi(\mathbf{x}_i^{r-1}), \mathbf{w}_{r-1} \rangle \mathbf{w}_{r-1}; \tag{3}$$

3 Terminate if the desired number of dimensions R has been reached (or $max\{||\phi(\mathbf{x}_i^r)|| : 1 \leq i \leq l\} < \epsilon$). Otherwise, increment r by 1 and go back to line 2.

Based on an initial training set, Algorithm 2 provides us with one or several orthogonal directions $\{\mathbf{w}_r | r = 1, \ldots, R\}$ which can be used as projectors to an R-dimensional subspace $\mathcal{S}$. Then each newly arrived instance $\mathbf{x}_t$ can be projected to $\mathcal{S}$ as $\langle \phi(\mathbf{x}_t), \mathbf{w}_r \rangle = \sum_{i=1}^{l} \alpha_i^r \kappa(\mathbf{x}_t, \mathbf{x}_i)$ for $r = 1, \ldots, R$. It is worth pointing out that all computations involved in RSVM can be based on kernel evaluation instead of the explicit $\phi(\mathbf{x}_t)$. From the second iteration, $\kappa(\mathbf{x}_i^r, \mathbf{x}_t^r)$ can be recursively computed by using (3) and $\kappa(\mathbf{x}_i^{r-1}, \mathbf{x}_t^{r-1})$, allowing different kernels to be adopted.

In this paper, we select $R = 1$ after some preliminary experiments. In fact, the assumption of $R = 1$ is realized by many classification models, starting from perceptrons, SVMs through to classification based on Gaussian Processes. All these models can be interpreted as imposing a single projection dimension where classification can be performed. Since such 1-dimensional projection directions are integral part of such classification machines, they are also good candidates for 1-dimensional subspaces on which to perform statistical test regarding concept drifts. Other supervised dimensionality reduction methods may also be used. However, techniques such as PCA are inherently unsupervised, and hence do not, by definition, satisfy our requirement for a low-dimensional subspace relevant to classification.

3.2 Class-Based Detection

While most existing detectors focus on detecting drifts by monitoring $P(Y|\mathbf{X})$ or $P(\mathbf{X})$, there has been a lack of attention to $P(\mathbf{X}|Y)$. Supervised information can be better utilized by class-conditional distributions because they focus on sub-regions of the input space. In HRDD, we suggest not only incorporating a distribution-based CDT to inspect data features from the perspective of marginal distribution, but also constructing one CDT for each class-conditional distribution $P(\mathbf{X}|Y = q)$, where $q \in \{0, 1, \ldots, Q\}$. The CDTs are initialized on its respective data stream (line 2, Algorithm 1). Note that only the marginal detector and one of the class-conditional detectors are activated each time an instance arrives.

Usually, the number of classes of a data stream is much lower than the number of dimensions. Therefore, HRDD is still expected to be computationally cheaper to implement than existing multivariate detectors that either try to estimate the distribution density or examine each dimension individually. By monitoring also the class-conditional distributions, HRDD captures both real and virtual drift, regardless of the effect on classification performance. Besides, since it synchronizes sub-regions of the input space, it is able to evaluate the effects on different classes and its detection sensitivity over smaller drifts is enhanced.

Different techniques can be chosen as the base CDT for this component. ICI-based CDT can be used as a reference example. A dominant advantage of this sequential CDT is that it is endowed with a refinement procedure that directly provides the estimated drift starting time T_{ref} [33]. Thus, a new dataset representing the most recent concept is automatically identified. For other drift detectors, the method introduced in [34] is recommended to identify T_{ref}.

Comparing with IV-Jac [24] which also monitors $P(\mathbf{X})$ and $P(\mathbf{X}|Y)$, we emphasize the following differences: a) our framework deals with continuous data features. IV-Jac cannot be directly applied to our problem setting; b) our framework can be used with various statistical CDTs and does not require prior knowledge about the drift to determine

TABLE 1
Construction of Retraining Sets After Detection

		Layer-I and II output from the Marginal Detector		
		Detected and Validated	Detected but Invalidated	No Detection
Layer-I and II output from Class 0 Conditional Detector	Detected and Validated	M: $TS^M = TS_C^M$ C0: $TS^0 = TS_C^0$ C1: $TS^1 = [TS^1, TS_C^1]$	M: $TS^M = TS_C^M$ C0: $TS^0 = TS0_C$ C1: $TS1 = [TS^1, TS_C^1]$	M: $TS^M = TS_C^M$ C0: $TS^0 = TS_C^0$ C1: $TS^1 = [TS^1, TS_C^1]$
	Detected but Invalidated	M: $TS^M = TS_C^M$ C0: $TS^0 = TS_C^0$ C1: $TS^1 = TS_C^1$	M: No retraining C0: No retraining C1: $TS^1 = [TS^1, TS_C^1]$	M: $TS^M = [TS^M, TS_C^M]$ C0: No retraining C1: $TS^1 = [TS^1, TS_C^1]$
	No Detection	M: $TS^M = TS_C^M$ C0: $TS^0 = TS_C^0$ C1: $TS^1 = TS_C^1$	M: No retraining C0: $TS^0 = [TS^0, TS_C^0]$ C1: $TS^1 = [TS^1, TS_C^1]$	

Without loss of generality, we assume the last instance received belongs to Class 0. Analogous definitions can be made for Class 1. TS^M, TS^0, TS^1 are the existing training sets for the marginal, Class 0 and Class 1 detectors, respectively. TS_C^M is composed of all instances representing the current concept in $[T_{ref}, \widehat{T}]$. TS_C^0 (TS_C^1) denotes the set of Class 1 (Class 0) instances in TS_C^M.

detection threshold; c) our approach works within a reduced feature subspace, hence is more robust against noise in the original data and scales better for high-dimensional data.

3.3 Knowledge Base Reconfiguration

Once a suspicious change is reported in the detection layer by at least one of the base detectors at time $\widehat{T}$, a potential drift starting time T_{ref} is estimated (lines 6-7, Algorithm 1). Then the validation layer is activated and an offline statistical test is used to compare the previous training set of the respective detector and instances from T_{ref} to $\widehat{T}$ to determine if the drift should be confirmed (line 8, Algorithm 1). If a drift is validated, detection time point $\widehat{T}$ is recorded. Afterwards, the existing HCDT framework discards all past data and reconfigure based on the most recent data only. This approach may be over-conservative for a supervised data stream as a drift may have uneven effects on different classes. Unnecessary rejection of data in a relatively stationary class leads to information loss, which can become problematic when available information is already scarce or expensive to obtain. Here we propose a novel and more flexible way of reconstructing the retraining sets in order to maintain as much useful information as possible for detector reconfiguration. The idea can be summarized as follows.

1) For data streams where we can confirm that a change has taken place (with a detected and validated change), the respective detectors are immediately reconfigured based on a latest dataset representing the current concept. It should be noted that when one class-conditional detector reports a validated change, it subsequently impacts the marginal distribution according to Equation (2), therefore in this case the marginal detector is also retrained.
2) For data streams where there is ambiguity if a change has taken place (a detected but invalidated change), we do not make any amendments to the existing detector.
3) For data streams where we are inclined to believe that no drift has taken place, all available and relevant instances are used as the new retraining set for the respective detectors. For instance, when a detection is reported by Class 0 detector but no validated detection from either the Class 1 detector or the marginal detector, we may combine the latest Class 1 instances in $[T_{ref}, \widehat{T}]$ with the previously existed Class 1 training set TS^1 to form a more informative retraining set. Hence, the performance of the detectors is expected to improve as extra relevant instances are used for retraining.

Hotelling T2 test has been shown to be a suitable complementary validation test for ICI-based CDT in the existing HCDT framework [21]. As a concrete realization under a bi-class scenario, the reconstruction scheme for all detectors after each detection can be summarized in Table 1. Based on the results from both the detection layer and validation layer, retraining datasets are constructed and the detectors are retrained accordingly (lines 11-12, Algorithm 1). Finally, all the validated changes are reported when there is no more data to arrive (line 13, Algorithm 1).

4 Computational Studies

This section presents four sets of experiments that evaluate the effectiveness and efficiency of HRDD. Experiment 1 aims to demonstrate the effectiveness of each component of the HRDD framework, especially when facing data streams with various dimensionalities. Experiment 2 illustrates the superiority of HRDD in drift detection on both real and virtual drifts over state-of-the-art methods. Experiment 3 validates that the superior performance provided by HRDD also benefits classification, even when integrated with a very simple classifier. Experiments 1-3 are based on datasets of synthetically generated sequences where the ground truth of drift occurrences is available. In Experiment 4, we demonstrate the role of HRDD on a real-world data stream. Finally, we provide a brief analysis on the computational time complexity of the approaches being considered in the experiments. All experiments were run on a CentOS 7.6 Computer with v4 2.20 GHz processor and 128 GB memory.

4.1 Performance Metrics

A variety of performance metrics for drift detection have been used in the literature. For instance, when counting the number of True Positive (TP), False Negative (FN) and False Positive (FP), some authors focus on if a detection is raised on a drifted sequence, instead of the number of detections raised [35], [36]. Differently, some authors pay attention to whether there are redundant detections after a TP, and distinguish between *Detected*, *Late*, *Missed* and *False* detections based on sliding windows [15], [37]. False detections before

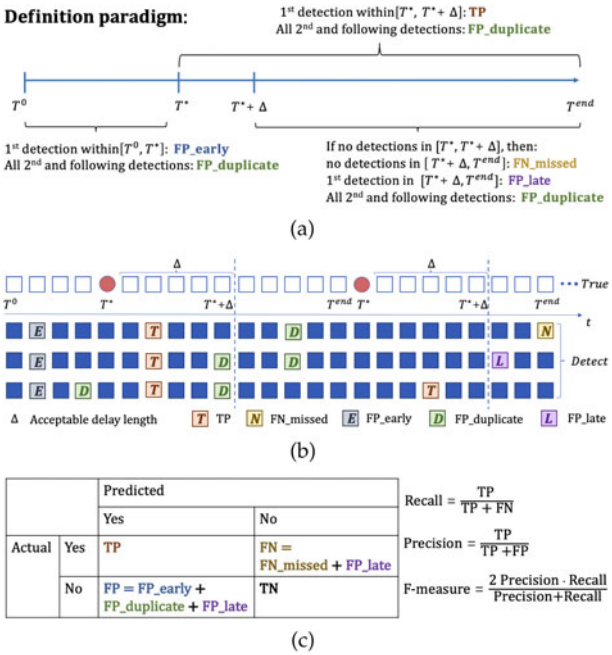

Fig. 3. Detection performance definition paradigm.

the first drifting point are neglected. In [9], all detections raised on a stream are taken into account and each single detection is categorized into TP or FP based on a specified window size. The notion of acceptable delay Δ was formally introduced in [38]. Here, FPs are defined as detections outside of the acceptable detection interval $[T^*, T^* + \Delta]$, but extra detections within the interval are neglected.

From a practical point of view, taking into account all detections raised on a stream is important. Distinguishing between various types of false alarms also helps to make targeted modifications. Therefore, when analysing the results of a reactive detector, we propose a more realistic and comprehensive definition paradigm as in Fig. 3a. Based on a predefined acceptable detection delay range $[T^*, T^* + \Delta]$ where T^* is the real drifting time, we define a TP as the first detection within this range, a FN_missed as a missed alarm throughout the concept. We also distinguish between three types of FPs: FP_early, FP_duplicate and FP_late. A FP_early is the first false alarm before T^* related to algorithm initialization, FP_duplicate's are redundant false alarms related to algorithm reconfiguration, and a FP_late is the first detection in $[T^* + \Delta, T^{end}]$ when there is no alarm raised in $[T^*, T^* + \Delta]$. An illustrative example is presented in Fig. 3b.

The total number of FPs and FNs are therefore FP = FP_early + FP_duplicate + FP_late and FN = FN_late + FP_late respectively. Performance of the detector is evaluated via number of TPs, FPs, FNs or Recall, Precision and F-measure as defined in Fig. 3c. For each synthetic dataset in the experiments, 30 sequences are generated, and all reported figures are summations (for TP and FP) or averages (for Recall, Precision, and F-Measure). Detection performance is measured for several acceptable lengths Δ = {500, 1000, 1500, 2000} so as to limit the maximum detection delay allowed.

4.2 Experimental Results

Experiment 1: Understanding HRDD. In order to better understand the novelty of HRDD relative to the existing hierarchical framework HCDT, we carry out a component-wise evaluation on data streams with varying dimensionalities. The characteristics of HRDD and several variations containing only partial components are presented in Table 2. HCDT-M is the existing HCDT framework which monitors the marginal input distribution only [21]. HCDT-CC is the existing HCDT framework applied to the class-conditional distributions. HDD is similar to HRDD in terms of inspection of both marginal and class-conditional distributions, but without projection to a reduced-space.

Synthetic data generated for this experiment is a set of d-dimensional moving hyperplanes $y = -a_0 + \sum_{i=1}^{d} a_i x_i$, $x_i \in [0, 1]$ and $y \in [0, d]$. This is a popular dataset in the field of drift detection [39], [40]. The generation mechanism also allows easy alteration of dataset dimensionality. To demonstrate the ability of HRDD to handle high-dimensional data, we considered $d = [5, 10, 15, 20, 30, 40]$. Data generation details can be found in Table 3. The data stream is balanced with 5% of class noise added. Each stream consists of 10,000 instances with one abrupt change at timestamp 5001.

Due to the page limit, we report only the results for Δ = 1000 and 2000 in Table 4. The following findings are also applicable to Δ = 500 and 1500. First, we notice that HRDD achieves the highest TP in almost all cases. This is true even for a tight Δ, indicating that HRDD can not only detect the drifts, but also detect them earlier than the existing HCDT and other variations being considered. Meanwhile, HRDD always reports the lowest FP. HDD, which is also based on this novel reconfiguration scheme but does not project data onto the low-dimensional space as HRDD does, always ranked second in terms of both TP and FP. In contrast, methods monitoring each dimension within the input space lead to much higher FP.

Comparing with the results of HCDT-CC, we can conclude that HRDD is very different from the existing HCDT applied on each class. The novelty of HRDD lies in not only class-based inspections, but also the projection of data onto

TABLE 2
Compared Detection Frameworks in Experiment 1

	detection			reconfiguration	
Algorithm	monitor $P(\mathbf{X})$	monitor $P(\mathbf{X}\|Y)$	reduced-space	single CDT	multiple CDTs
HRDD	✓	✓	✓		✓
HCDT-M (HCDT)	✓			✓	
HCDT-CC		✓		✓	
HDD	✓	✓			✓

TABLE 3
Synthetic Data Generation of d-Dimensional Hyperplane Datasets

Concept	$d-dimensional\ hyperplane$
1	$a_0^1 = -1.5;\quad a_i^1 = i \times 0.1\ \forall\, i \in \{1, ..., d\}$
2	$a_0^2 = a_0^1 - 1;\quad a_i^2 = a_i^1 - 0.5\ \forall\, i \in \{1, ..., d\}$

TABLE 4
Detection Performance on Data Streams With Increasing Dimensionality

	Δ =	1000												2000											
		TP						FP						TP						FP					
	Γ =	2.25	2.5	2.75	3.0	3.25	3.5	2.25	2.5	2.75	3.0	3.25	3.5	2.25	2.5	2.75	3.0	3.25	3.5	2.25	2.5	2.75	3.0	3.25	3.5
5D	HRDD	**30**	**27**	**22**	**16**	11	7	**18**	**10**	**16**	**15**	19	23	**30**	**30**	**30**	**30**	**30**	**30**	18	7	8	**1**	**0**	**0**
	HCDT-M	5	2	3	0	0	0	31	27	27	26	21	**18**	24	23	24	11	8	4	**12**	**6**	**6**	15	13	14
	HCDT-CC	26	23	19	14	11	**9**	22	17	17	17	21	22	**30**	**30**	**30**	**30**	**30**	**30**	18	10	**6**	**1**	2	1
	HDD	23	25	19	**16**	**12**	**9**	19	11	18	16	**18**	22	29	26	29	**30**	**30**	**30**	13	10	**8**	**2**	**0**	1
10D	HRDD	**30**	**30**	28	**27**	**24**	**21**	**21**	**12**	**10**	**9**	**9**	**11**	**30**	**30**	**30**	**30**	**30**	**30**	**21**	**12**	**8**	**6**	3	**2**
	HCDT-M	27	27	26	21	20	18	157	66	35	13	10	17	28	29	**30**	28	29	29	156	64	31	6	**1**	6
	HCDT-CC	29	28	**29**	25	20	13	130	83	57	56	47	53	**30**	**30**	**30**	**30**	**30**	**30**	129	81	56	51	37	36
	HDD	25	28	28	25	**24**	**21**	114	81	59	21	24	15	27	29	**30**	29	29	**30**	112	80	57	17	19	6
15D	HRDD	**30**	**30**	**30**	**30**	**29**	**27**	**11**	**12**	**9**	8	**4**	**4**	**30**	**30**	**30**	**30**	**30**	**30**	**11**	**12**	**9**	8	**3**	1
	HCDT-M	23	27	28	28	28	24	344	203	83	**4**	16	6	27	29	29	29	**30**	**30**	340	201	82	**3**	14	**0**
	HCDT-CC	22	23	26	21	18	12	134	106	82	60	58	61	25	25	**30**	28	29	29	131	104	78	53	47	44
	HDD	21	25	28	29	28	**27**	216	176	119	40	40	13	27	28	**30**	29	**30**	**30**	210	173	117	40	38	10
20D	HRDD	**30**	**30**	**30**	**30**	**30**	**28**	**15**	**10**	**9**	**9**	**9**	7	**30**	**30**	**30**	**30**	**30**	**30**	**15**	**10**	**9**	**9**	**9**	5
	HCDT-M	19	21	24	26	28	27	405	211	123	40	18	**3**	23	23	26	26	29	**30**	401	209	121	40	17	**0**
	HCDT-CC	10	13	9	16	10	14	107	95	56	36	38	30	16	20	13	19	17	24	101	88	52	33	31	20
	HDD	17	19	18	20	22	27	238	164	96	68	58	22	23	21	21	22	25	29	232	162	93	66	55	20
30D	HRDD	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**31**	**28**	**28**	**22**	**12**	**30**	**30**	**30**	**30**	**30**	**30**	**29**	**31**	**28**	**28**	**22**	**12**
	HCDT-M	29	25	22	16	11	12	1052	781	506	361	242	132	**30**	**30**	**30**	**30**	**30**	29	1051	776	498	347	223	115
	HCDT-CC	**30**	**30**	**30**	**30**	**30**	**30**	288	213	164	123	97	86	**30**	**30**	**30**	**30**	**30**	**30**	288	213	164	123	97	86
	HDD	**30**	**30**	29	29	**30**	**30**	517	342	214	155	95	76	**30**	**30**	**30**	**30**	**30**	**30**	517	342	213	154	95	76
40D	HRDD	**30**	**30**	**30**	**30**	**29**	**29**	**21**	**14**	**10**	**5**	**4**	**3**	**30**	**30**	**30**	**30**	**30**	**30**	**21**	**14**	**10**	**5**	**3**	**2**
	HCDT-M	**30**	27	20	16	12	7	1304	986	697	489	305	256	**30**	29	**30**	26	23	15	1304	988	687	479	294	248
	HCDT-CC	**30**	**30**	**30**	**30**	**29**	**29**	353	234	162	109	68	64	**30**	**30**	**30**	**30**	**30**	**30**	353	234	162	109	67	63
	HDD	**30**	**30**	**30**	**30**	**29**	**29**	601	407	328	217	139	69	**30**	**30**	**30**	**30**	**30**	**30**	601	407	328	217	138	68

Methods with high TP and low FP are preferred. Best results given the specified parameter Γ and acceptable delay length Δ are in bold. Cases where HRDD achieves the best result among all the methods being compared are underlined.

the reduced-space, and the utilization of both marginal and class-conditional information for reconfiguration. As dimensionality increases, the superiority of HRDD becomes more dominant, confirming its ability to operate efficiently even for high-dimensional data streams. Also, comparing the performance presented in Table 4 horizontally, it can be seen that HRDD is relatively insensitive to the parameter of the base detector, making it a more reliable and stable approach among the compared methods.

Experiment 2: Drift Detection Ability. In this section we aim to compare the drift detection ability of HRDD on a wide range of drifts with the latest hierarchical detection methods, HCDT [21] and HLFR [23] introduced in Section 1. These consolidated frameworks have already been shown to perform better than their individual base detector counterparts. We also compare HRDD with two classic performance-based benchmarks, DDM [6] and EDDM [7], which have not been used as base change detectors in the above-mentioned frameworks. Since the detection result from performance-based detectors is contingent on the choice of classifier, two classifiers are adopted: an SVM and a decision tree. All hyper-parameters of the comparative algorithms were taken directly from the original papers. The setting of HRDD follows the experimental setting for HCDT (Experiment B in [21]). Detection Recall, Precision, and F-measure are reported for $\Delta = \{500, 1000, 1500, 2000\}$.

In this experiment we first test on data streams with one abrupt drift only. With drift affecting $P(Y|\mathbf{X})$ or not and its magnitude being small or large, there are 4 possible scenarios for a single drift. These cases will be examined individually. Afterwards, data streams with multiple drifts are used for testing. The following synthetic datasets are generated for this experiment:

1) *4D Multivariate Gaussian* (Fig. 4): This dataset contains sequences with one drift only. We synthetically generate drifts affecting the target concept differently by changing one class-conditional distribution independently. Possible drift scenarios are visualized in Fig. 4. In order to reflect the 4 scenarios, 4 subsets of 4D Multivariate Gaussian streams are generated. Each data stream consists of 10,000 observations, and a single abrupt change takes place at 5001. The magnitude of drift is controlled by the change in within-class distance d_w. The effect on the target concept is controlled by the change in between-class distance d_b. The (d_w, d_b) pair for the initial concept is always (0,0). For

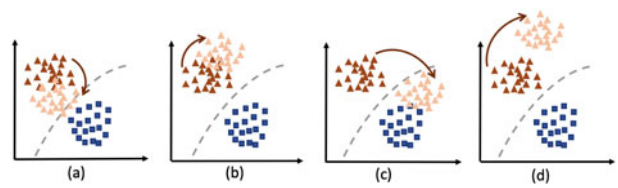

Fig. 4. Illustration of various drift types of 4D Multivariate Gaussian. (a) small drift affecting $P(Y|\mathbf{X})$; (b) small drift not affecting $P(Y|\mathbf{X})$; (c) large drift affecting $P(Y|\mathbf{X})$; (d) large drift not affecting $P(Y|\mathbf{X})$. Data generation details are given in Table 5.

TABLE 5
Synthetic Data Generation of 4D Multivariate Gaussian

Concept	$4D\_Gaussian$			
	(a)	(b)	(c)	(d)
1	$\boldsymbol{\mu}^1_{C_0} = [0,0,0,0]$ $\boldsymbol{\mu}^1_{C_1} = [0.8,0.8,0.8,0.8]$ $\boldsymbol{\Sigma}^1_{C_0} = \boldsymbol{\Sigma}^1_{C_1} = \mathbb{1}_4$			
2	$\boldsymbol{\mu}^2_{C_0} = [-0.2, 0.1, -0.2, 0.1]$	$\boldsymbol{\mu}^2_{C_0} = [-0.2, -0.2, -0.2, -0.2]$	$\boldsymbol{\mu}^2_{C_0} = [0.4, -0.3, 0.4, 0.4]$	$\boldsymbol{\mu}^2_{C_0} = [-0.3, -0.4, -0.4, 0.4]$
	$\boldsymbol{\mu}^2_{C_1} = \boldsymbol{\mu}^1_{C_1}$ $\boldsymbol{\Sigma}^2_{C_0} = \boldsymbol{\Sigma}^2_{C_1} = \mathbb{1}_4 + 0.2 \times (\mathbb{J}_4 - \mathbb{1}_4)$			

The illustration is given in Fig. 4.

TABLE 6
Synthetic Data Generation of 6D Multivariate Gaussian Datasets

Concept	$6D\_Gaussian$	Concept	$6D\_Gaussian$
1	$\boldsymbol{\mu}^1_{C_0} = [2,2,3,3,4,4]$ $\boldsymbol{\mu}^1_{C_1} = [1,1,2,2,3,3]$; $\boldsymbol{\Sigma}^1_{C_0} = \boldsymbol{\Sigma}^1_{C_1} = \mathbb{1}_6$	4	$\boldsymbol{\mu}^1_{C_0} = [2.8,2.8,3.4,3.4,3.8,3.8]$ $\boldsymbol{\mu}^4_{C_1} = \boldsymbol{\mu}^3_{C_1}$; $\boldsymbol{\Sigma}^4_{C_0} = \boldsymbol{\Sigma}^4_{C_1} = \mathbb{1}_6$
2	$\boldsymbol{\mu}^1_{C_0} = [2.6,2.6,3.8,3.8,4.2,4.2]$ $\boldsymbol{\mu}^2_{C_1} = \boldsymbol{\mu}^1_{C_1}$; $\boldsymbol{\Sigma}^2_{C_0} = \boldsymbol{\Sigma}^2_{C_1} = \mathbb{1}_6$	5	$\boldsymbol{\mu}^1_{C_0} = [2.6,2.6,3.0,3.0,3.4,3.4]$ $\boldsymbol{\mu}^5_{C_1} = \boldsymbol{\mu}^4_{C_1}$; $\boldsymbol{\Sigma}^5_{C_0} = \boldsymbol{\Sigma}^5_{C_1} = \mathbb{1}_6$
3	$\boldsymbol{\mu}^1_{C_0} = [2.2,2.2,3.4,3.4,4.4,4.4]$ $\boldsymbol{\mu}^3_{C_1} = \boldsymbol{\mu}^2_{C_1}$; $\boldsymbol{\Sigma}^3_{C_0} = \boldsymbol{\Sigma}^3_{C_1} = \mathbb{1}_6$		

The illustration is given in Fig. 5.

scenarios (a-d) in Fig. 4, the (d_w, d_b) pair are set to (0.5, -0.9), (0.5, 0.8), (1.0, -0.8), and (1.0, 1.1), respectively. Data generation details can be found in Table 5.

2) *6D Multivariate Gaussian* (Fig. 5): This dataset contains multiple-drift streams. We consider a scenario where a series of drifts is not detrimental to classification at the beginning, but eventually impairs the accuracy after several evolutions. A simple illustration of this situation is presented in Fig. 5. Each sequence is of length 25,000 and contains 5 concepts. The evolution of concept can be summarized as the (d_w, d_b) pair being (0, 0), (0.4, 2.4), (0.4, 1.9), (0.4, -1.4), and (0.4, -2.1) for each drift. Details of the data generation process can be found in Table 6.

3) *Rotating Checkerboard* (Fig. 6): In this multiple-drift benchmark dataset [41], all 4 drifts lead to a strong change in classification boundary. Each stream is of length 25,000 and contains 5 concepts. Examples are sampled uniformly from the unit square with a dimensionality of 2, and the labels are set by a checkerboard with 0.5 tile width. At each concept drift, the checkerboard is rotated by an angle of $\pi/6$ radians.

Detection performance for *4D Multivariate Gaussian* is summarized in Fig. 7. Overall, HRDD ranked first in 14 out of the 16 cases (4 datasets and 4 Δ's) in terms of F-measure, indicating its ability to achieve the best trade-off between recall and precision. Performance-based detectors HLFR, EDDM and DDM only secure high recall values for real drifts affecting $P(Y|\mathbf{X})$, which cause an evident degradation in classification accuracy (Figs. 7a and 7c). For drifts not harming classification performance, i.e., drifts not affecting $P(Y|\mathbf{X})$ (Figs. 7b and 7d), performance-based detectors fail and the distribution-based detector HCDT becomes the second best detector after HRDD in terms of detection F-measure. In addition, HRDD also surpasses HCDT by a great amount when drift magnitude is small (Figs. 7a and 7b). This is due to the fact that the detection mechanism monitoring class-conditional distributions makes HRDD more sensitive to even a lightest change in the overall input space. For drifts with greater magnitude (Figs. 7c and 7d), the performance of HCDT improves, but it still falls behind HRDD in all but one case.

Table 7 presents how many times each individual detector is activated among all 30 TP detections. When a drift is caused by Class 0 only, the class-conditional distribution of Class 0 and the marginal distribution are both affected. Results in Table 7 shows that 28 out of the 30 drifts can be captured by the respective detector or the marginal detector, which comes in line with our expectation. For scenarios b) and d), Class 0 moves away from Class 1, leading to a relatively greater change in the marginal distribution comparing with scenarios a) and c), hence these two scenarios result in more activations of the marginal detector. This also demonstrates that the combination of both marginal and class-conditional inspections in HRDD is indeed helpful.

Moving to the multiple-drift scenarios, HRDD also outperforms its competitors in all 8 cases in terms of F-measure as shown in Figs. 8 and 9. For *6D Multivariate Gaussian* (Fig. 8), since the magnitude of each single drift is relatively small, HCDT requires two or more consecutive drifts in order for the effect of the drift series to be sufficiently noticeable on the marginal distribution. Performance-based detectors HLFR, EDDM and DDM are only able to detect the last

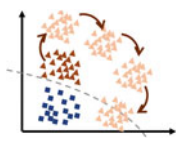

Fig. 5. Illustration of 6D Multivariate Gaussian. Data generation details are given in Table 6.

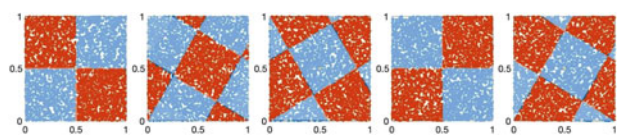

Fig. 6. Illustration of rotating checkerboard.

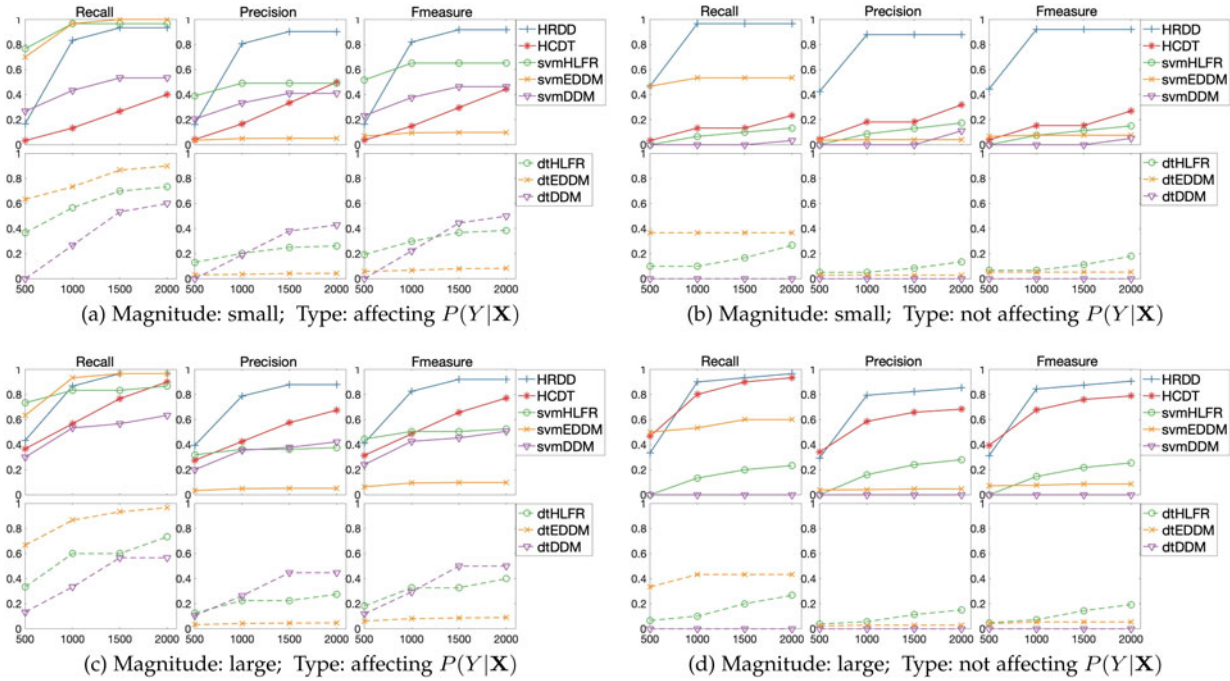

Fig. 7. Detection performance for 4D Multivariate Gaussian against acceptable delay lengths. Subfigures (a-d) correspond to scenarios (a-d) in Fig. 4, respectively.

one or two drifts in Fig. 5, since earlier drifts do not deteriorate classification performance.

On the *Rotating Checkerboard* dataset, the effectiveness of HRDD can also be clearly identified in Fig. 9. As expected, HCDT does not perform well because purely distribution-based detectors fail to detect changes affecting the labelling mechanism only [42]. The distribution of overall input space of this dataset remains unchanged. This phenomenon demonstrates that detecting concept drift by monitoring the class-conditional distributions is helpful. For this dataset, $P(Y|\mathbf{X})$ is significantly affected by all drifts, allowing the performance-based detectors to capture the drifts more acutely. Therefore HLFR, EDDM and DDM achieved very high recall values. However, the precision plot reveals that significantly more false alarms are triggered. Therefore, HRDD, which secures the highest F-measure, is still the most reliable choice. Another interesting finding from Figs. 8 and 9 is that when a decision tree is used as the base classifier, HLFR and EDDM achieve much better than when an SVM classifier is used. This confirms that the choice of classifier plays an important role in performance-based drift detection. In contrast, the performance achieved by HRDD is irrelevant to the base classifier.

TABLE 7
Number of Activations of Each CDT for 4D Multivariate Gaussian

Drift type	Marginal	Class 0	Class 1
a)	7	21	2
b)	11	17	2
c)	7	21	2
d)	10	18	2

Drift types are shown in Fig. 4. Class 0 is the drifted class. Total number of activations is 30.

Based on the above analysis, it can be concluded that for real drifts affecting $P(Y|\mathbf{X})$, HRDD performs no worse, and in many cases better than existing performance-based detectors. For virtual drifts not directly affecting $P(Y|\mathbf{X})$, HRDD performs better than both distribution-based and performance-based detectors. HRDD also performs particularly better than the comparative methods when the changes have minor effect on the overall input distribution.

Experiment 3: Role in Classification. The focus of this paper is to propose a new drift detection framework HRDD. Intuitively, accurate detection and localisation of drifts would help to improve classification because it leads to just-in-time model-retraining. What classification model and retraining techniques achieve the lowest classification error in a reactive streaming environment is a matter for future work. However, in order to evaluate the role of a more accurate and efficient detector in streaming data classification environments, we present the prequential classification error[1] at the end of each sub-concept for *6D Multivariate Gaussian* and *Rotating Chekerboard* datasets. For performance-based detectors (HLFR, EDDM and DDM), the classifier is always retrained on a fixed-length recent window. For distribution-based detectors (HCDT and HRDD), a simple detect-then-retrain technique is adopted. Instances from the estimated drift starting point T_{ref} to detection point $\widehat{T}$ are used as the retraining set. Experiments are carried out with two classifiers, SVM and decision tree.

Tables 8 and 9 demonstrate that HRDD helps to achieve a lower classification error on both datasets no matter which base classifiers is adopted. For the *6D Multivariate Gaussian* dataset, recall that the first two drifts are virtual. The

1. As in [43], the prequential error at timestamp i is defined as $E_i = \frac{S_i}{B_i} = \frac{L(y_i,\hat{y}_i)+\lambda S_{i-1}}{1+\lambda B_{i-1}}$ where $L(y_i,\hat{y}_i)$ is the 0-1 classification loss function, $S_1 = L_1$, $B_1 = 1$ and λ is a decay factor set to 0.999).

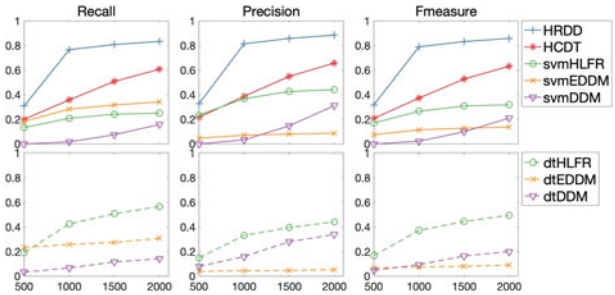

Fig. 8. Detection performance for 6D Multivariate Gaussian. For detectors HLFR, EDDM and DDM: Linear SVM as the base classifier (top); decision tree as the base classifier (bottom).

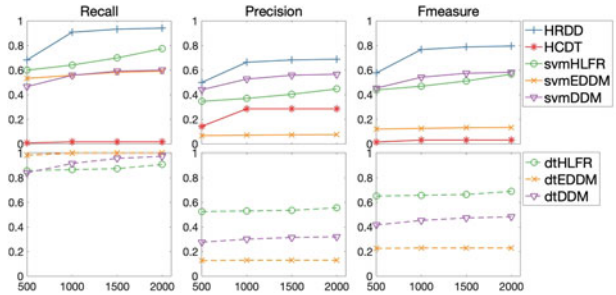

Fig. 9. Detection performance for Rotating Checkerboard. For detectors HLFR, EDDM and DDM: RBF SVM as the base classifier (top); decision tree as the base classifier (bottom).

classification task actually becomes easier as the classes move further away from each other. Performance-based detectors consider these drifts to be irrelevant and do not detect such drifts. Even in these cases, HRDD, which accurately detects all types of drifts, leads to an even lower error than the performance-based detectors. This supports the hypothesis that when the optimal decision boundary has shifted but performance is not deteriorated, retraining the classifier can still be beneficial. For the *Rotating Chekerboard* dataset, recall that the drifts affect the labelling mechanism only. The data distribution-based detector HCDT fails to make accurate detections, hence the much worse classification performance than the performance-based detectors.

Overall, HRDD can help in reducing classification error regardless of the drift type. For both real and virtual drifts, incorporating HRDD in a classification model can achieve a lower or at least comparable classification error than both performance-based and distribution-based detectors.

Experiment 4: Real-World Scenarios. In the above experiments, synthetic data streams are used to better understand the functionality, efficiency, and effectiveness of HRDD. For real-world data streams, there is no ground truth regarding the existence or location of drifts. Therefore, the performance metrics used for synthetic datasets cannot be employed. Here we report the number of detections and prequential classification error to compare the methods. A classification system that achieves the lowest number of detections as well as the lowest classification error is preferred.

Electricity [44] is a dataset collected from the Australian New South Wales Electricity Market. It contains 45,312 instances and each example is described by 8 features. The class label identifies the change of the price relative to a moving average of the last 24 hours. (i.e., up and down). We note that there has been a dispute regarding the usage of this dataset for concept drift detection analysis due to the temporal correlation within the data [45]. Nonetheless, it is still one of

TABLE 8
Classification Error for 6D Multivariate Gaussian

	SVM-linear					Decision Tree				
Concept	HRDD	HCDT	HLFR	EDDM	DDM	HRDD	HCDT	HLFR	EDDM	DDM
1	**0.12** (0.01)	**0.12** (0.01)	0.13 (0.02)	0.19 (0.05)	**0.12** (0.01)	**0.23** (0.02)	0.24 (0.02)	0.24 (0.02)	0.29 (0.05)	**0.23** (0.02)
2	**0.06** (0.01)	0.07 (0.01)	0.07 (0.01)	0.09 (0.04)	0.07 (0.01)	**0.15** (0.01)	0.17 (0.02)	0.17 (0.02)	0.19 (0.05)	0.17 (0.02)
3	**0.04** (0.01)	0.05 (0.01)	0.06 (0.01)	0.07 (0.02)	0.06 (0.01)	**0.11** (0.02)	0.13 (0.02)	0.15 (0.02)	0.16 (0.04)	0.15 (0.02)
4	**0.05** (0.01)	**0.05** (0.01)	0.07 (0.02)	0.08 (0.02)	0.07 (0.01)	**0.11** (0.01)	0.12 (0.02)	0.14 (0.02)	0.19 (0.03)	0.17 (0.02)
5	**0.09** (0.01)	0.11 (0.01)	0.10 (0.02)	0.15 (0.05)	0.10 (0.02)	**0.16** (0.02)	0.19 (0.02)	0.18 (0.03)	0.25 (0.04)	0.17 (0.02)

Average prequential classification error (standard deviation in parenthesis) at the end of each sub-concept is presented. Best results are in bold. Cases where HRDD achieves the best result among all the methods being compared are underlined.

TABLE 9
Classification Error for Rotating Checkerboard

	SVM-rbf					Decision Tree				
Concept	HRDD	HCDT	HLFR	EDDM	DDM	HRDD	HCDT	HLFR	EDDM	DDM
1	**0.12** (0.04)	**0.12** (0.04)	**0.12** (0.05)	0.21 (0.05)	0.14 (0.04)	**0.03** (0.02)	0.04 (0.02)	0.04 (0.03)	0.04 (0.03)	0.05 (0.02)
2	**0.42** (0.04)	0.49 (0.02)	0.46 (0.04)	0.47 (0.04)	0.44 (0.05)	**0.21** (0.06)	0.49 (0.06)	0.25 (0.09)	0.26 (0.06)	0.23 (0.11)
3	**0.13** (0.10)	0.84 (0.16)	0.27 (0.17)	0.31 (0.16)	0.31 (0.15)	0.07 (0.09)	0.90 (0.20)	**0.05** (0.05)	**0.05** (0.06)	0.13 (0.15)
4	**0.43** (0.09)	0.52 (0.07)	0.48 (0.03)	0.48 (0.05)	0.47 (0.07)	**0.21** (0.12)	0.53 (0.10)	0.26 (0.13)	0.25 (0.08)	0.22 (0.09)
5	**0.11** (0.07)	0.16 (0.16)	0.35 (0.20)	0.37 (0.15)	0.33 (0.16)	**0.05** (0.07)	0.09 (0.20)	0.09 (0.14)	**0.05** (0.05)	0.10 (0.13)

Average prequential classification error (standard deviation in parenthesis) at the end of each sub-concept is presented. Best results are in bold. Cases where HRDD achieves the best result among all the methods being compared are underlined.

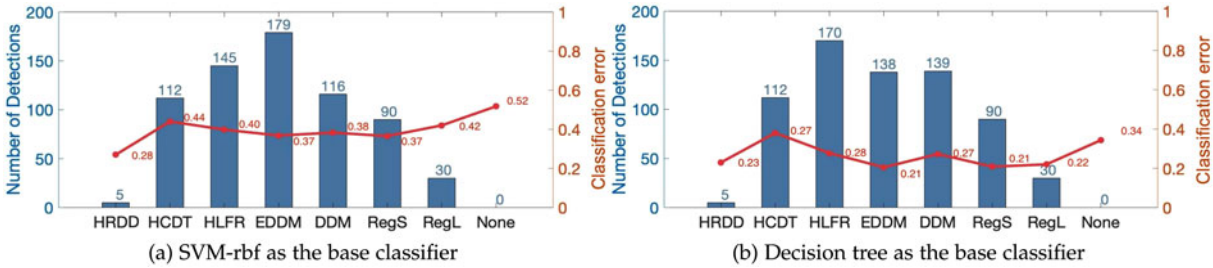

Fig. 10. Detection and classification performance for the electricity data stream. The bar plot represents the number of detections raised by each method, and the line plot records the prequential classification error at the end of the data stream.

the most commonly used real-world data streams in this area of research [46]. In order to mitigate this issue, we also compare with situations where regular retraining takes place and where no detector is adopted.

The number of detections and the classification error obtained by the methods concerned are presented in Fig. 10. $RegS$ and $RegL$ stand for regular retraining with small and large intervals (500 and 1500), respectively. The selection of interval length directly affects classification performance as well as the computational cost. $None$ indicates the no-detector situation. From the line plot representing the classification error, it can be seen that adding a drift detector always help reducing the classification error since all methods lead to lower error when comparing with the no-detector situation.

From the bar plot representing the number of detections, it can be seen that HRDD always ranked first, with only 5 detections regardless of the choice of base classifier. In contrast, its competitors all raise many more detections, causing higher overhead cost. HRDD not only bears a very low computational burden from retraining, but also helps to maintain a satisfactory classification performance. The line plots in Fig. 10 demonstrate that when an SVM is used as the base classifier, the error obtained by HRDD ranked first, and is much lower than what has been achieved by its competitors. When a decision tree is adopted, HRDD ranked fourth with a classification error of 0.23, being only 0.02 higher than the best result achieved by EDDM. Examining the number of detections and classification error together, we may conclude that in summary, HRDD achieves the best trade-off between classification performance and computational cost on this real-world data stream.

4.3 Time Complexity Analysis

DDM [6] and EDDM [7] have a constant time complexity ($\mathcal{O}(1)$) at each time point, since they monitor a single error-rate based statistic. Although the base detector LFR [10] in HLFR [23] also has complexity ($\mathcal{O}(1)$), the validation layer requires extra training of P classifiers (P=1000 in the original paper). Assuming $\mathcal{O}(K)$ is the computational complexity of training a new classifier, the time complexity for HLFR is $\mathcal{O}(KP)$, which is usually much higher than ($\mathcal{O}(1)$).

TABLE 10
Average Runtime for Each Reported Detection (s.)

Dataset	HRDD	HCDT	HLFR	EDDM	DDM
4D Gaussian	0.2496	0.1044	49.0637	0.2645	2.1117
6D Gaussian	0.6129	0.2989	63.3693	0.5735	3.0722
Rotating Checkerboard	0.2004	0.4088	36.2308	1.1619	6.9635

HCDT [21] adopts an univariate test on each dimension in the detection layer and one offline test in the validation layer. If the complexity of the base detector is ($\mathcal{O}(1)$), the complexity of the overall framework is ($\mathcal{O}(d)$) where d is the dimensionality of input space. HRDD has a similar structure but adopts a univariate test on each dimension of the reduced-space for each class in the detection layer. The time complexity is ($\mathcal{O}(rQ)$) (r = 1 and Q = 2 in this paper so ($\mathcal{O}(rQ)$) is close to ($\mathcal{O}(1)$)).

For multivariate data streams of higher dimensionality, the advantage of HRDD will become more significant since the number of classes is usually much lower than the number of dimensions. The average runtime for a reported detection is summarized in Table 10. It is worth noting that performance-based detectors generally have longer runtime since they also include a classifier training procedure which data distribution-based detectors do not. Practitioners should take this into consideration when choosing the appropriate detector depending on the application scenario.

5 Conclusion

We have proposed a data distribution-driven and class-based hierarchical drift detection framework HRDD for multivariate supervised data streams. The proposed framework first maps the data to a lower dimensional subspace, and then detects drifts in that space relevant to the given classification task. It utilizes information from both marginal distribution and class-conditional distributions of the supervised data stream. Based on the effect of drift on each class, a novel reconfiguration scheme aiming to maintain as many as possible relevant instances for retraining is incorporated within the algorithm. HRDD detects both real and virtual drifts, regardless of their effects on classification. It is also capable of detecting subtle drifts which can hardly be captured by existing distribution-based detectors. HRDD is computationally light and efficient when operating on higher-dimensional data streams. The proposed approach outperformed others by achieving a better recall-precision trade-off within the given acceptable delay length when compared with the latest distribution-based and performance-based methods in the literature.

In the current work, we only projected the data onto a single-dimensional linear subspace. How various characteristics of the subspace would affect the detection performance should be further investigated in future work. In fact, HRDD can also be used with many different settings. How various combinations of detection and validation tests will affect the performance on different data streams is

another topic worth further exploration. Besides, extension to accommodate multi-class data streams and even imbalanced-class data streams are also future research directions following this current work.

ACKNOWLEDGEMENTS

This work was supported in part by the Research Institute of Trustworthy Autonomous Systems, in part by the Guangdong Provincial Key Laboratory under Grant 2020B121201001, in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant 2017ZT07X386, and in part by Shenzhen Science and Technology Program under Grant KQTD2016112514355531. Peter Tino was supported by the European Commission Horizon 2020 Innovative Training Network SUNDIAL, Project ID: 721463.

REFERENCES

[1] A. Tsymbal, "The problem of concept drift: Definitions and related work," *Comput. Sci. Dept., Trinity College Dublin*, vol. 106, no. 2, pp. 58–64, 2004.

[2] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363, Dec. 2019.

[3] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, "A survey on data preprocessing for data stream mining: Current status and future directions," *Neurocomputing*, vol. 239, pp. 39–57, 2017.

[4] S. Wang, L. L. Minku, and X. Yao, "A systematic study of online class imbalance learning with concept drift," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4802–4821, Oct. 2018.

[5] H. Hu, M. Kantardzic, and T. S. Sethi, "No free lunch theorem for concept drift detection in streaming data classification: A review," *Wiley Interdiscipl. Rev. Data Mining Knowl. Discov.*, vol. 10, no. 2, 2020, Art. no. e1327.

[6] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Proc. Braz. Symp. Artif. Intell.*, 2004, pp. 286–295.

[7] M. B.-García, J. del C.-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, "Early drift detection method," in *Proc. 4th ECML PKDD Int. Workshop Knowl. Discov. Data Streams*, 2006, pp. 77–86.

[8] G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand, "Exponentially weighted moving average charts for detecting concept drift," *Pattern Recognit. Lett.*, vol. 33, no. 2, pp. 191–198, 2012.

[9] M. Harel, S. Mannor, R. El-Yaniv, and K. Crammer, "Concept drift detection through resampling," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1009–1017.

[10] H. Wang and Z. Abraham, "Concept drift detection for streaming data," in *Proc. Int. Joint Conf. Neural Netw.*, 2015, pp. 1–9.

[11] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[12] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi, "An information-theoretic approach to detecting changes in multi-dimensional data streams," in *Proc. Symp. Interface Statist. Comput. Sci. Appl.*, 2006, pp. 1–24.

[13] C. Alippi, G. Boracchi, and M. Roveri, "Change detection tests using the ICI rule," in *Proc. Int. Joint Conf. Neural Netw.*, 2010, pp. 1–7.

[14] G. Ditzler and R. Polikar, "Hellinger distance based drift detection for nonstationary environments," in *Proc. IEEE Symp. Comput. Intell. Dyn. Uncertain Environ.*, 2011, pp. 41–48.

[15] N. Lu, G. Zhang, and J. Lu, "Concept drift detection via competence models," *Artif. Intell.*, vol. 209, pp. 11–28, 2014.

[16] L. Bu, D. Zhao, and C. Alippi, "An incremental change detection test based on density difference estimation," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 47, no. 10, pp. 2714–2726, Oct. 2017.

[17] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surveys*, vol. 46, no. 4, pp. 1–37, 2014.

[18] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool, "Seeking the strongest rigid detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3666–3673.

[19] W. J. Faithfull, J. J. Rodríguez, and L. I. Kuncheva, "Combining univariate approaches for ensemble change detection in multivariate data," *Inf. Fusion*, vol. 45, pp. 202–214, 2019.

[20] P. Sobolewski and M. Wozniak, "Concept drift detection and model selection with simulated recurrence and ensembles of statistical detectors," *J. Universal Comput. Sci.*, vol. 19, no. 4, pp. 462–483, 2013.

[21] C. Alippi, G. Boracchi, and M. Roveri, "Hierarchical change-detection tests," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 246–258, Feb. 2017.

[22] W. Härdle and L. Simar, *Applied Multivariate Statistical Analysis*. Berlion, Germany: Springer, 2007, vol. 22007.

[23] S. Yu, Z. Abraham, H. Wang, M. Shah, Y. Wei, and J. C. Príncipe, "Concept drift detection and adaptation with hierarchical hypothesis testing," *J. Franklin Inst.*, vol. 356, no. 5, pp. 3187–3215, 2019.

[24] Y. Zhang, G. Chu, P. Li, X. Hu, and X. Wu, "Three-layer concept drifting detection in text data streams," *Neurocomputing*, vol. 260, pp. 393–403, 2017.

[25] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, "Characterizing concept drift," *Data Mining Knowl. Discov.*, vol. 30, no. 4, pp. 964–994, 2016.

[26] J. Gao, W. Fan, J. Han, and P. S. Yu, "A general framework for mining concept-drifting data streams with skewed distributions," in *Proc. SIAM Int. Conf. Data Mining*, 2007, pp. 3–14.

[27] Q. Tao, D. Chu, and J. Wang, "Recursive support vector machines for dimensionality reduction," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 189–193, Jan. 2008.

[28] V. Vapnik, "Pattern recognition using generalized portrait method," *Autom. Remote Control*, vol. 24, pp. 774–780, 1963.

[29] D. A. Blythe, P. Von Bunau, F. C. Meinecke, and K.-R. Muller, "Feature extraction for change-point detection using stationary subspace analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 631–643, Apr. 2012.

[30] C. Wu, B. Du, and L. Zhang, "A subspace-based change detection method for hyperspectral images," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 815–830, Apr. 2013.

[31] Y. Kawahara, T. Yairi, and K. Machida, "Change-point detection in time-series data based on subspace identification," in *Proc. 7th IEEE Int. Conf. Data Mining*, 2007, pp. 559–564.

[32] Y. Jiao, Y. Chen, and Y. Gu, "Subspace change-point detection: A new model and solution," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 6, pp. 1224–1239, Dec. 2018.

[33] C. Alippi, G. Boracchi, and M. Roveri, "Adaptive classifiers with ICI-based adaptive knowledge base management," in Proc. *Int. Conf. Artif. Neural Netw.*, 2010, pp. 458–467.

[34] H. V. Poor, *Detection of Abrupt Changes: Theory and Application. Michèle and Igor V. Nikiforov*, Englewood Cliffs, NJ, USA: Prentice Hall, 1996, vol. 32, no. 8.

[35] C. Alippi, G. Boracchi, and M. Roveri, "A hierarchical, nonparametric, sequential change-detection test," in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 2889–2896.

[36] Y. Kim and C. H. Park, "An efficient concept drift detection method for streaming data under limited labeling," *IEICE Trans. Inf. Syst.*, vol. 100, no. 10, pp. 2537–2546, 2017.

[37] F. Gu, G. Zhang, J. Lu, and C.-T. Lin, "Concept drift detection based on equal density estimation," in *Proc. Int. Joint Conf. Neural Netw.*, 2016, pp. 24–30.

[38] A. Pesaranghader and H. L. Viktor, "Fast hoeffding drift detection method for evolving data streams," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2016, pp. 96–111.

[39] L. L. Minku, A. P. White, and X. Yao, "The impact of diversity on online ensemble learning in the presence of concept drift," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 5, pp. 730–742, May 2010.

[40] N. Lu, J. Lu, G. Zhang, and R. L. De Mantaras, "A concept drift-tolerant case-base editing technique," *Artif. Intell.*, vol. 230, pp. 108–133, 2016.

[41] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1517–1531, Oct. 2011.

[42] C. Alippi, G. Boracchi, and M. Roveri, "Just-in-time classifiers for recurrent concepts," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 4, pp. 620–634, Apr. 2013.
[43] J. Gama, R. Sebastião, and P. P. Rodrigues, "Issues in evaluation of stream learning algorithms," in *Proc. 15th ACM SIGKDDO Int. Conf. Knowl. Discov. Data Mining*, 2009, pp. 329–338.
[44] M. Harries and N. S. Wales, "Splice-2 comparative evaluation: Electricity pricing," *Artif. Intell. Group, Sch. Comput. Sci. Eng., Univ. New South Wales*, Kensington, NSW, Australia, Tech. Rep., 1999.
[45] I. Zliobaite, "How good is the electricity benchmark for evaluating concept drift adaptation," 2013, *arXiv:1301.3524*.
[46] A. Bifet, J. Read, I. Žliobaitė, B. Pfahringer, and G. Holmes, "Pitfalls in benchmarking data stream classification and how to avoid them," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2013, pp. 465–479.

Shuyi Zhang received the bachelor's degree in mathematics and the master's degree in statistics from the Imperial College of London, U.K., in 2015 and 2016, respectively. She is currently working toward the PhD degree in computer science with the University of Birmingham, U.K., under co-supervision with the Southern University of Science and Technology, China. Her research interests include machine learning, concept drift detection, imbalanced classification, and online learning.

Peter Tino received the MSc degree from the Slovak University of Technology and the PhD degree from the Slovak Academy of Sciences. Since 2003, he has been with the School of Computer Science, University of Birmingham, Edgbaston, Birmingham, U.K., where he is currently a full professor and the chair in complex and adaptive systems. His current research interests include dynamical systems, machine learning, probabilistic modelling of structured data, evolutionary computation, and fractal analysis. He was the recipient of the Fulbright Fellowship in 1994, the U.K.–Hong-Kong Fellowship for Excellence in 2008, three Outstanding Paper of the Year Awards from the *IEEE Transactions on Neural Networks* in 1998 and 2011 and the *IEEE Transactions on Evolutionary Computation* in 2010, and the Best Paper Award at ICANN 2002. He is currently on the editorial boards of several journals. He was a fulbright fellow with NEC Research Institute, Princeton, NJ, USA and a postdoctoral fellow with the Austrian Research Institute for AI, Vienna, Austria and with Aston University, Birmingham, U.K.

Xin Yao (Fellow, IEEE) received the BSc degree from the University of Science and Technology of China (USTC) in 1982, the MSc degree from the North China Institute of Computing Technologies, Beijing, in 1985, and the PhD degree from USTC, Hefei, in 1990. He is currently a chair professor and the founding head of computer science with the Southern University of Science and Technology, Shenzhen, China, and a part-time professor of computer science with the University of Birmingham, UK. His major research interests include evolutionary computation and ensemble machine learning, online learning with concept drift and class imbalance learning, and their applications to fault diagnosis. He was a distinguished lecturer with IEEE Computational Intelligence Society. He was the president from 2014 to 2015 of IEEE Computational Intelligence Society and the editor-in-chief from 2003 to 2008 of *IEEE Transactions on Evolutionary Computation*. He was the recipient of the 2001 IEEE Donald G. Fink Prize Paper Award, 2010, 2016, and 2017 IEEE Transactions on Evolutionary Computation Outstanding Paper awards, 2011 IEEE Transactions on Neural Networks Outstanding Paper Award, 2010 BT Gordon Radley Award for Best Author of Innovation (Finalist), several best paper awards at conferences, a 2012 Royal Society Wolfson Research Merit Award, the 2013 IEEE CIS Evolutionary Computation Pioneer Award, and the 2020 IEEE Frank Rosenblatt Award.