Digit. Signal Process. 182 (2026) 106325

Contents lists available at ScienceDirect

Digital Signal Processing

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

Concept drift detection in delayed and partially labeled data streams: An experimental survey

Brenno Alencar [a], Guilherme Cassales [b], Heitor Murilo Gomes [c], Cássio Prazeres [a], Tatiane Rios [a], Albert Bifet [b], Ricardo Rios [a,*]

[a] Institute of Computing Federal University of Bahia, Salvador, 40170-110, Bahia, Brazil
[b] Artificial Intelligence Institute University of Waikato, Hamilton, 3240, Waikato, New Zealand
[c] School of Engineering and Computer Science Victoria University of Wellington, Wellington, 6140, New Zealand

ARTICLE INFO

Keywords:
Data stream
Concept drift
Delayed labeling
Partially labeled

ABSTRACT

Fully labeled data is the ideal scenario for supervised model training. However, labels might be scarce in many situations, specially when large data is produced as an open-ended stream. Moreover, in dynamic streams, it is assumed that the underlying process generating the data is non-stationary, changing its behavior over time. Therefore, concepts learned by the classification model are likely to change, and non-adaptive models will suffer performance degradation, a phenomenon known as concept drift. The challenge of recognizing and reacting to such changes is even more complex by facing streams with partial or delayed labels. This refers to realistic scenarios where the true label of an incoming point is not immediately available (delay) or might never be available (partial). In this work, we focus on investigating the performance of concept drift detectors in handling delayed and partially labeled data streams. We categorize the main methods from the literature, providing a taxonomy of concept drift detectors and an overview of the most influential approaches for evaluating these detectors under conditions of delayed or partial labeling. Finally, we conduct a series of experiments to analyze the performance of these detectors in scenarios with label scarcity. Concluding, the survey discusses its main limitations and offers insights into potential avenues for future research in the field.

1. Introduction

Training models on fully-labeled data represent the optimal scenario for traditional supervised learning approaches. Moreover, such approaches assume that the observed data is in a state of statistical equilibrium, meaning that its properties remain unchanged over time. In other words, the probability distribution of newly collected observations is assumed to be the same as that used during the training phase. However, real-world systems often present two major challenges that impact learning performance: dynamic behaviors over time and limited label availability. Dynamic behaviors change not only the system states but also the associated labels, such as consumers' behaviors, opinions about products or political parties, and network attacks. When changes affect the posterior probabilities of class labels, thereby the output distribution and influencing subsequent predictions, this phenomenon is known as Concept Drift (CD) [1]. As discussed by [1], most algorithms designed to detect CD are devoted to identifying univariate streams of correct/incorrect predictions, requiring the availability of labeled data as promptly as possible.

When facing missing or delayed labels, the learning capacity of such algorithms tends to decrease. In this context, if the ground truth is not immediately available, the algorithm executions do not guarantee expected performance. This issue is even more threatening in applications based on data streams, which are potentially unbounded in size, often requiring real-time processing and analysis. The search for solutions to this challenge has motivated a new research area devoted to dealing with Delayed Partially Labeled Data Streams (DPLDS) [2].

The impact of label latency on the performance of learning models in data streams is a widely discussed topic in the literature [3]. Recently, [2] presented a comprehensive survey identifying change-adaptive approaches that predominantly leverage semi-supervised learning in scenarios involving delayed labeling within data streams. These studies, therefore, underscore the importance of investigating delayed and partially-labeled data streams.

In recent years, several surveys have explored concept drift and challenges related to semi-supervised learning and non-stationary data streams, particularly under conditions of label scarcity and delay [1–3]. However, there remains a notable gap in the literature regarding

* Corresponding author.
E-mail addresses: brenno.mello@ufba.br (B. Alencar), guilherme.cassales@waikato.ac.nz (G. Cassales), heitor.gomes@vuw.ac.nz (H.M. Gomes), prazeres@ufba.br (C. Prazeres), tatianerios@ufba.br (T. Rios), abifet@waikato.ac.nz (A. Bifet), ricardoar@ufba.br (R. Rios).

https://doi.org/10.1016/j.dsp.2026.106325

Available online 8 June 2026

Table 1
Comparison between our work and related surveys.

Survey	Main Focus	Label Availability	Experimental Benchmark
Gama et al.(2014) [1]	Concept drift	Assumes immediate label availability	No
Gomes et al.(2022) [2]	Semi-supervised learning for delayed and partially labeled data streams	Focus on semi-supervised learning under label delay	Yes - illustrative experiments
Fahy et al.(2022) [3]	Label scarcity in non-stationary data streams	Focus on learning under limited labeling budgets	No
This work	**Concept drift detection under delayed and partially available labels**	**Investigates impact of missing labels and delay on drift detection**	**Yes - extensive empirical evaluation under realistic label availability conditions**

concept drift detection methods specifically designed for data streams that are both delayed and partially labeled. Existing surveys typically address these topics from a broad perspective or focus on semi-supervised learning in general, without a dedicated emphasis on drift detection under these constraints.

To address this gap, our work combines both survey-based and experimental contributions. First, we conducted a comprehensive literature review to identify state-of-the-art drift detectors applicable to settings with delayed and limited label availability. From this review, we developed a novel taxonomy tailored to the specific challenges of concept drift detection in such scenarios. Importantly, this taxonomy distinguishes itself from prior surveys by structuring the problem space with a focus on the implications of label availability and delay on detection strategies.

Table 1 highlights key differences between our work and previous surveys. While Gama et al. (2014) [1] focus on traditional concept drift approaches, Gomes et al. (2022) [2] presents a broad overview of semi-supervised learning in delayed data streams, and Fahy et al. (2022) [3] focuses on learning under label scarcity in non-stationary environments, our work provides an experimental survey focus on concept drift detection under delayed and partially available labels. Additionally, unlike these surveys, we complement the conceptual framework with an extensive experimental study, designed to assess detector performance under varying levels of label delay and different types of drift. Together, the proposed taxonomy and empirical analysis offer a distinct and complementary perspective to existing literature, strengthening the understanding of concept drift in realistic, partially supervised streaming environments. Our results show which approaches handle label delay and scarcity better, and point to promising directions for future research.

In summary, the main contributions of this work are as follows:

- This work organizes the current literature on concept drift detection in delayed and partially labeled data streams, allowing the reader to comprehend the state of the art and paving the way for future research;
- A challenging aspect of dealing with delayed and partially labeled data streams is ensuring a fair evaluation of drift detectors. To address this issue, we comprehensively discuss evaluation procedures, thus supporting researchers to better state and assess their studies;
- Besides providing a literature review, we also evaluated the effectiveness of the detectors discussed in this manuscript. We conducted experiments to analyze their performance in data streams with partial and delayed labels. Additionally, we carried out experiments including both abrupt and gradual concept drift using seven different types of synthetic data streams.

The remaining of this manuscript is organized as follows: we first introduce the problem statement, clearly identifying similarities and differences with related problems in Section 2; subsequently, in Section 3, we introduce existing concept drift methods for DPLDS streaming data; Section 4 includes a discussion regarding the evaluation strategies in CD detection on delay and partially labeled data streams; in the final Section 6, we draw some conclusions and discuss new insights for future research, according to the authors' perceptions.

2. Problem definition

A data stream S can be defined as a sequence of labeled examples (x_i, y_i), such that $i = \{1, 2, \dots\}$ represents time instants, and $x_i \in X$ is a collection of data points with their respective labels $y_i \in Y$. The majority of concept drift detection algorithms are formulated under the assumption that the true label y_i of each incoming data stream instance is available immediately after prediction. Accordingly, both drift detection and performance monitoring are conducted under the implicit assumption of negligible or nonexistent label delay. However, this assumption cannot be assured once label unavailability may often occur due to transmission failure, data acquisition error, or lack of specialists.

The relationship between the presence of labels and their availability can be formally expressed as a temporal-mapping function $T(\cdot)$ that precisely extracts discrete time unit t when x_i and y_i are available. Based on this function, Gomes et al. [2] define four different definitions describing the relationships between data and labels:

1. **Immediate and fully labeled:** $\forall x_i \in X,\ \forall y_i \in Y,\ \Delta_{x_i} = T(y_i) - T(x_i) = 1$. The latency validation between x_i and y_i corresponds exactly to a one-time unit.
2. **Delayed and fully labeled:** $\forall x_i \in X,\ \forall y_i \in Y,\ \Delta_{x_i} = T(y_i) - T(x_i) = L$, where L is a random variable that represents the discrete delay between x_i and y_i. The discrete delay is limited by the finite range $L \in \mathbb{Z}_+$.
3. **Immediate and partially labeled:** Similarly to 1, the latency validation is a one-time unit. However, only partially x_i has a corresponding entry on Y. In that case, the sequence of labeled examples is $\exists x_i \in X,\ \nexists y_i \in Y,\ (x_i, ?),\ \Delta_{x_i} = T(y_i) - T(x_i) = \infty$.
4. **Delayed and partially labeled:** Based on 2 and 3, some labels are delayed, and others are missing (infinitely delayed).

In addition to considering label availability, discussing whether the data distribution is stationary or evolving is essential in our problem definition. By default, our assumption leans towards evolving data distributions, which leads to the occurrence of concept drifts, impacting decision boundaries.

In summary, the concept drift area studies changes in the conditional distribution of the output from a target variable concerning the input. Although we can estimate changes in advance for specific real-world situations, concept drifts are generally characterized as unexpected and unpredictable [1]. Formally, the concept drift between two points at time $t = 0$ and time $t = 1$ can be defined by equation 1, in which P_0 is the coexisting distribution of the input variables X and the targets Y at time $t = 0$.

$$\exists X : P_0(X, Y) \neq P_1(X, Y), \tag{1}$$

Table 2
Different Concept Drift scenarios based on changes in $P(Y|X)$ and $P(X)$ [6].

		$P(Y\|X)$ Change	No Change
$P(X)$	Change	(1) Observable	(2) Observable
	No Change	(3) Unobservable	(4) No Drift

Briefly, modifications in the prior probabilities of classes $P(Y)$ and changes in their conditional probabilities $P(X|Y)$ can lead to drifts in their posterior probabilities $P(Y|X)$. As discussed by Gama et al. (2014) [1], drifts can be understood as changes in data behavior that occur from different sources: (i) sudden/abrupt, when there is a change from one concept to another; (ii) incremental, when the behavior alternates among several intermediate concepts; (iii) gradual, when data behavior slowly changes over time; and (iv) reoccurring, when previous concepts are detected likewise.

Although we adopt the widely used taxonomy of concept drift [1], it is important to clarify the criteria underlying its categorization. In the literature, drift types are primarily distinguished according to the evolution of the relationship between features and the target and the speed of change [4]. Abrupt drift corresponds to a rapid transition from one data distribution to another within a short time interval. In contrast, gradual drift occurs when this transition unfolds progressively, during which instances from both the old and the new distributions may be observed simultaneously. Incremental drift, on the other hand, is characterized by a very slow and continuous evolution of the underlying concept, such that consecutive instances may exhibit no statistically significant differences, while substantial changes emerge only over longer time horizons. Finally, recurrent concept drift, as explained by Gunasekara et al. [5], refers to the reappearance of previously observed data distributions over time, the transitions can be abrupt, gradual, incremental, partial, or evolving. Moreover, concept recurrence may occur with different temporal patterns. Periodic recurrent drifts exhibit regular and predictable recurrences over time, e.g., seasonal phenomena. In contrast, semi-periodic recurrent drifts arise when only a subset of concepts recurs periodically, and others reappear irregularly. In random recurrent drifts, concept reoccurrence follows no discernible temporal pattern, making recurrence difficult to anticipate from the data stream alone [5]. It is important to highlight that these changes have a direct impact on the overall prediction and can be categorized into the following types:

- Real Concept Drift: modifications in $P(Y|X)$, in which changes occur with or without a simultaneous shift in $P(X)$. Real concept drift is alternatively named concept shift or conditional change.
- Virtual Drift: changes in distributions of incoming data, $P(X)$, without influencing $P(Y|X)$.

In online classification, identifying actual concept drifts is based on changes in the posterior probabilities of labels, denoted as $P(Y|X)$, thus requiring adjustments in the decision rule. However, when real-time labels are unavailable, concept drift can be discerned through changes in the data distribution, $P(X)$. As discussed by Žliobaite (2010) [6], different scenarios are derived from the combination of such changes, as illustrated in Table 2. In the first scenarios, (1) and (2), drifts are observable in case of changes in $P(X)$. On the other hand, when just $P(Y|X)$ changes, the drifts are unobservable, as shown in Scenario (3). Finally, no drift is considered if there is no change in $P(Y|X)$ and $P(X)$, as illustrated in Scenario (4).

In this work, we focus on the state of the art of concept drift detection in scenarios with delayed and partially labeled data streams. To illustrate this scenario, consider Fig. 1, which shows a data stream produced over time following a probability distribution $P(X)$. The initial distribution, represented in green, yields the data x_0, x_1, and x_2. Their respective labels follow the joint distribution $P(X,Y)$. The first data represents an immediate label, once $T(y_0) - T(x_0) = 1$. For simplicity, the one-time unit is represented in our figure by $\Delta = 1$. Although immediate feedback is not always feasible in real-world applications, this setting is commonly adopted in the literature for evaluating learning algorithms and drift detectors. It enables an upper-bound performance assessment by assuming that labels become available without delay, thus avoiding the complexities introduced by label latency.

A concept drift caused by changes in $P(X)$ and/or $P(X,Y)$ is illustrated as a different color (red). After this change, one may notice the arrival of a delayed label y_1 associated to x_1, which was produced from the previous distribution, representing a temporal-mapping function $T(y_1) - T(x_1) = L$, i.e., $\Delta > 1$. The data stream is also considered partially labeled when given labels are missing (infinitely delayed), as illustrated by the data x_2, in which its respective label is unavailable, i.e., $T(y_2) - T(x_2) = \infty$.

In Fig. 1, we still represent a reoccurring concept drift, in which current data behavior (in red) is replaced by the previous one (in green), representing a new change in $P(X)$ and/or $P(X,Y)$. Similarly, we also illustrate other temporal-mapping examples: (i) immediate-labeled data as $T(y_i) - T(x_i) = 1$ ($\Delta = 1$); (ii) partially-labeled data as $T(y_{i+1}) - T(x_{i+1}) = \infty$; and (iii) data with delayed label as $T(y_{i+n}) - T(x_{i+n}) = L$, i.e., $\Delta > 1$. We also illustrate with y_{i+2} the availability of a delayed label produced from a previous distribution (concept).

To illustrate the impact of delayed label, we created a data stream with 4 abrupt concept drifts (at instances 20,000, 40,000, 60,000, and 80,000). This experiment followed a change-adaptation methodology in which base models were reset whenever a drift was detected. The methods used to create the data (AGRAWAL) and detect drifts (Hoeffding Tree and ADWIN) will be detailed in the following sections. Fig. 2(a) and (b) show the number of detected drifts (y-axis) over the processing of 100,000 instances, both with and without delayed labels. The results clearly indicate that label delay has a substantial impact on the performance of the drift detection algorithm.

In addition to analyzing concept drift detections, Fig. 3(a) and (b) illustrate the effect of label delays on the model's accuracy over time. The comparison shows that delayed labeling not only reduces overall accuracy but also prolongs the recovery period after each drift, highlighting the significant challenges delays introduce in maintaining model performance in evolving data streams.

3. Concept drift detectors

Most concept drift detectors are fundamentally designed to monitor changes in data distributions. In their simplest form, these detectors track how distributions evolve over time. Many supervised detectors rely on the availability of true labels, and are typically developed and evaluated without explicitly accounting for delays in label arrival. However, real-world applications often involve delayed or partially labeled data, where the detector no longer has a consistent or reliable view of this distribution. This lack of immediate feedback complicates drift detection, requiring more robust and adaptable approaches. Therefore, using detectors that require immediate labeling may be impractical, highlighting the need for effective label-independent drift detectors capable of operating without labeled data. In this section, we organized the most well-known detectors according to the taxonomy shown in Fig. 4.

The taxonomy starts separating the detectors into two main scenarios according to the availability of labels: (i) immediate and (ii) delayed/partially. Traditionally, research focused on immediate labels encompasses sequential analysis, control charts, two-distribution monitoring, and contextual methods. In turn, delayed and partial labels are investigated by monitoring raw data, estimating class labels, and classifying outputs. In the following sections, we explore these scenarios.

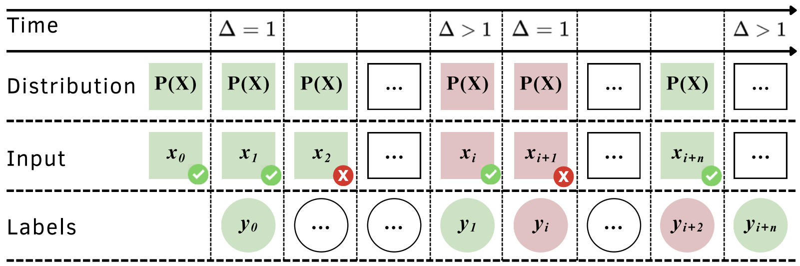

Fig. 1. Example of a data stream with different label availabilities and a reoccurring concept drift.

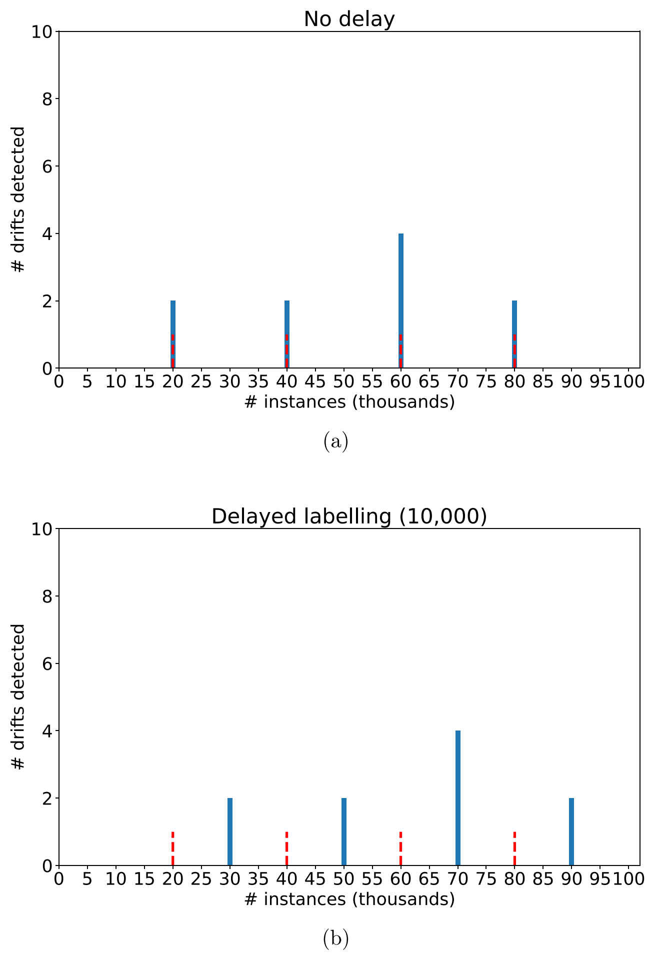

Fig. 2. Concept drifts detected in data streams with (a) no label delay and (b) delayed labels. Blue bars represent detected drifts, while red-dotted vertical lines mark the actual drift points.

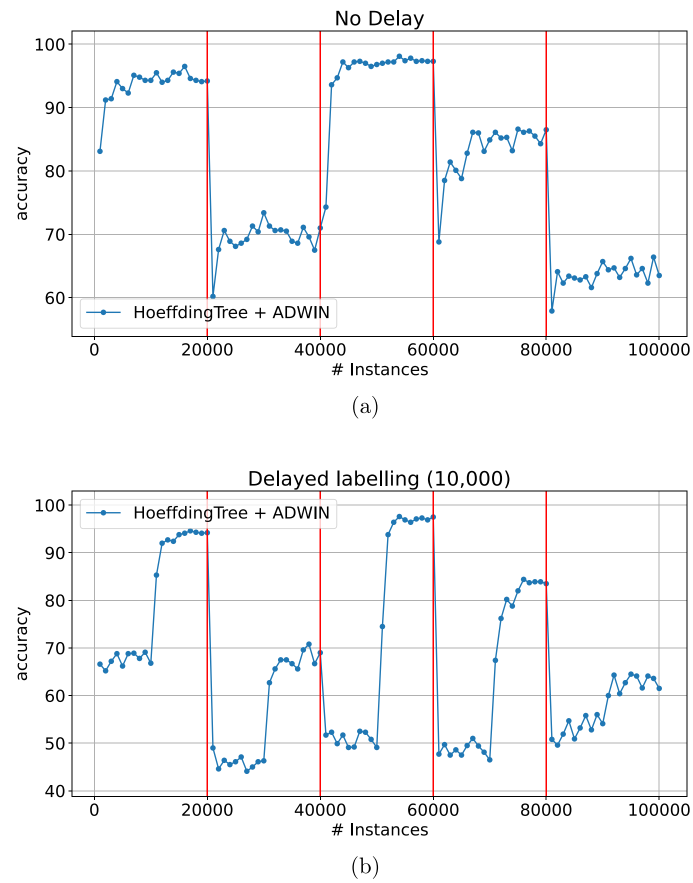

Fig. 3. Model accuracy in scenarios (a) without label delay (81.12%) and (b) with label delay (66.33%). Red lines indicate the occurrence of concept drifts.

3.1. *Detectors with immediate labelling*

Concept drift detectors are generally designed to handle data streams where labels are promptly and fully available, meaning all labels are immediately accessible. Detectors using immediate labeling are based on the error e_t, computed from the difference between the expected label y_t and the predicted one $\hat{y}_t$. In this context, Gama et al. (2014) [1] presented a comprehensive survey, categorizing the literature based on the strategies used to identify and quantify changes as they occur. In this section, we have summarized the discussion about these strategies on three distinct pathways: *Sequential Analysis*, *Control Chart*, and *Distributions Monitoring*.

3.1.1. *Sequential analysis*

Detectors based on *Sequential Analysis* typically leverage the Sequential Probability Ratio Test (SPRT) to identify the exact drift point. Formally, SPRT can be defined as follows: Let X_n be a sequence of examples, where the subset of examples X_w, with $1 < w < n$, is generated from an unknown distribution P_0, and the subset X_n^w is generated from another unknown distribution P_1. When the underlying distribution transitions from P_0 to P_1 at point w, the probability of observing subsequences under P_1 is expected to be significantly higher than that under P_0. This change in probability serves as the basis for detecting concept drift.

Cumulative Sum (CUSUM) [7] is a sequential method based on SPRT, in which the drift test is computed by: $g_0 = 0$ and $g_t = \max(0, g_{t-1} + (e_t - \delta))$. The change is detected when $g_t > \tau$, such that δ corresponds to the magnitude of the acceptable changes, and τ is a user-defined limit. As one may notice, the drift is detected when values in the sequential analyses positively increase. To detect changes as negative trends, the method uses $\min(\cdot)$, instead, and the change is detected

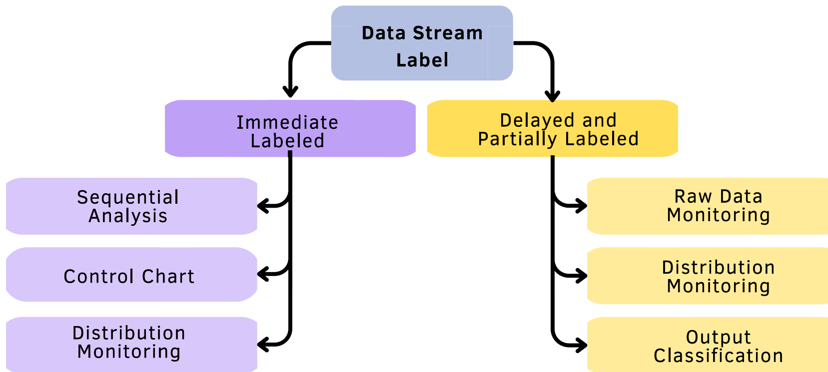

Fig. 4. A taxonomy of concept drift detection on delay and partially labeled data stream.

when g_t is lower than a negative limit. In this method, considered memoryless, the general idea is to identify changes when the average of the input data is significantly different from zero.

Page-Hinkley test (PHT) [7] is also a sequential method designed to detect changes in the average of a Gaussian signal. Its test considers a cumulative variable U_n computed by $U_n = \sum_{t=1}^{n}(e_t - \bar{x}_n - \delta)$, in which $\bar{x} = \frac{1}{n}\sum_{t=1}^{n} e_t$ and δ correspond to the allowed magnitude of changes. To detect changes, PHT calculates the minimum value $m_n = \min(U_t)$, $t = \{1, \dots, n\}$, and monitors the difference $PHT = U_n - m_n$. Concept drift is detected when PHT exceeds a predefined threshold τ, i.e., $PHT > \tau$, where τ controls the false alarm rate. Increasing τ reduces false alarms but may result in missed or delayed drift detections. Thus, tuning τ represents a trade-off between reducing false alarms and minimizing the delay in detecting concept drift.

3.1.2. Control chart

Control Chart, also known as *Statistical Process Control* (SPC), comprises standard statistical techniques used to monitor and regulate the quality of a product throughout continuous manufacturing processes. Drift detectors can use SPC to measure the rate of changes in data streams as discussed in the following methods.

The Exponentially Weighted Moving Average (EWMA) [8] method is used to detect increases in the mean of a sequence of random variables. In summary, EWMA computes $Z_t = (1-\lambda)Z_{t-1} + \lambda e_t$, $t = \{1, \dots, n\}$, in which $Z_0 = \mu_0$, and e_t is a stream of prediction error, independent and identically distributed random variables with means μ_0 and μ_1 before and after the change points, respectively. The λ parameter controls the influence of the most recent data in comparison with the data previously analyzed. Therefore, it is possible to understand that, regardless of the distribution of the data stream X, the mean $\mu(\cdot)$ and standard deviation $\sigma(\cdot)$ of Z_t are defined in $\sigma_{Z_t} = \sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2_t})}\sigma_X$, such that and $\mu_{Z_t} = \mu_t$.

While a change is not detected ($\mu_0 = \mu_t$), the estimator Z_t evolves around the expected mean and standard deviation. A change leads the value of μ_t to μ_1, and Z_t reacts by diverging from μ_0 towards μ_1, thus triggering an alert as stated in $Z_t > \mu_0 + \tau\sigma_{Z_t}$. Parameter τ, called control limit, determines when Z_t must differentiate from μ_0 for a change to be signaled. The value of τ is usually chosen to ensure that the detector obtains some predefined level of performance.

3.1.3. Distribution monitoring

Detectors based on *Distribution Monitoring* work by creating two time windows: a fixed one used as reference summarizing past data and a sliding one responsible for capturing recent examples. Statistical tests are then applied to compare the distributions from both. If the tests indicate the distributions are likely different (signaling a shift in the underlying process), a change is triggered at the beginning of the sliding window.

ADaptive sliding WINdow (ADWIN) [9] is the most well-known detector based on *Distribution monitoring*. In the ADWIN algorithm, as each error data point e_t arrives, a fixed-size detection window W slides forward to include the newest data. The algorithm compares sub-windows within W to determine if their averages differ significantly, concluding that a change in the underlying data distribution has occurred. ADWIN leverages the Hoeffding bound to assess whether the average difference between two sub-windows is statistically significant.

Another distribution-based detector is DDM (Drift Detection Method) [10], which uses the binomial distribution to calculate the overall probability of error occurrences within a sample of n instances. At each sampled point x_i in the sequence, the error rate is defined as the probability of misclassification p_i, with a standard deviation represented by $s_i = \sqrt{p_i(1-p_i)/i}$. It is assumed that as the number of examples increases, the error rate of the learning algorithm (p_i) will decrease if the distribution of examples remains stable. A notable uptick in the algorithm's error rate indicates a potential shift in the class distribution, implying that the current decision model may be inadequate.

Early Drift Detection Method (EDDM) [11] is an extended version of DDM, which was formulated to enhance the detection of gradual concept drift while maintaining robust performance in the face of abrupt changes. Its fundamental approach involves assessing the distance between two error classifications rather than solely focusing on error counts. As the learning method progresses, it refines its predictions, expanding the gap between successive errors.

Hoeffding's Inequality-based Drift Detection Method (HDDM) [12] is a statistical approach to detect changes in the data distribution. It is based on Hoeffding's Inequality, which provides an upper bound on the probability that the sample mean of independent and identically distributed random variables deviates from the actual mean by more than a certain amount. The basic idea is to maintain a sliding window of recent samples and calculate the error rate for each window. If the error rate exceeds a certain threshold, it indicates a potential drift in the data distribution.

Statistical Test of Equal Proportions [13] (STEPD) was designed to identify changes in data patterns rapidly and precisely. Its core concept revolves around evaluating two types of accuracy: recent and overall. This method is based on two key assumptions. First, if the target concept remains static, the classifier's accuracy on the most recent W examples will reflect the overall accuracy of the learning process. Second, a significant reduction in recent accuracy suggests a drift in the underlying concept. Finally, another relevant work is presented by Žliobaite (2010) [6], where hypothesis tests are proposed for comparing two samples using univariate statistical methods: the two-sample *t*-test, Kolmogorov-Smirnov test, and Wilcoxon rank-sum test. In binary classification tasks, only a single sequence is typically analyzed. However, in multi-class classification, multiple sequences must be examined.

3.2. Detectors with partially and/or delay labeled

The detectors presented in this section were selected due to their performance in analyzing delayed and partially labeled data streams, as discussed in Section 2 – Definition 4. In summary, they are organized in the right branch of the taxonomy presented in Fig. 4, also using three different pathways: *Raw Data Monitoring, Distribution Monitoring*, and *Output Classification*.

In order to achieve a deeper understanding of the selected classifiers, we categorize and discuss them within three groups: partially label-dependent, label-independent, and delayed label classifiers. Partially label-dependent detectors require instance labels to execute, though some instances may lack labels. In contrast, label-independent detectors operate without relying on any labels. Lastly, delayed label detectors focus on process data streams where labels become available after a delay, which could potentially be indefinite.

3.2.1. Raw data monitoring

Souza et al. (2020) [14] proposed a label-independent detector, called Image-Based Drift Detector (IBDD), that uses a visual representation to monitor changes in data distributions over time. IBDD employs two data windows (w_1 and w_2) of equal size to detect drifts. w_1 works as a reference window, storing the data from a known and stable concept, while w_2 is a sliding window that updates continuously with each new example from the stream. The d-dimensional data from w_1 and w_2 are then transformed into two distinct 2-dimensional grayscale images. Concept Drift is identified by analyzing pixel differences rather than comparing the differences between data distributions for each feature individually.

The study published by Klikowski (2022) [15] presented the Centroid Distance Drift Detector (CDDD) algorithm to detect drifts in data streams by analyzing distances in data chunks. CDDD comes in two variants: partially label-dependent and label-independent. The method examines the inter-chunk dynamics by evaluating centroid distance matrices. Variations in these distances mean changes in the underlying data distribution. Notably, the label-independent variant eliminates the need for object labels, assessing centroid distances across the entire data chunk without class-based partitioning.

3.2.2. Distribution monitoring

Faithfull et al. (2019) [16] introduced an ensemble combination strategy, where each ensemble model monitors a distinct feature of the input space. The approach, also considered label-independent, employs a straightforward voting mechanism and a threshold to detect drifts.

Other studies suggested employing the Kolmogorov-Smirnov test for concept drift detection [17–19]. Based on this suggestion, Dos Reis et al. (2016) [20] propose the Incremental Kolmogorov-Smirnov (IKS) algorithm to detect concept drifts in an online setting. IKS relies solely on data from the feature space, making it unaffected by delayed labeling or label scarcity. As a result, IKS is classified as label-independent.

Similarly following a label-independent strategy, the Discriminative Drift Detector (D3) method, proposed by Gözüaçık et al. (2019) [21], utilizes a discriminative classifier over a sliding window to monitor changes in data distribution. The classifier is periodically trained and tested to determine whether new samples originate from a distribution similar to the current one. The data from the training windows are organized such that the oldest members within a window are labeled as 'old' and assigned a value of 0. The remaining members are labeled as 'new' and assigned a value of 1. A logistic regression model is then trained as a discriminative classifier to distinguish between the 'old' and 'new' labels. D3 uses the Area Under the Curve (AUC) metric to assess separability, indicating the model's ability to distinguish between classes. A low AUC, approaching 0.5, suggests significant overlap in class distributions, pointing to a suboptimal model. Based on the degree of class divergence, D3 detects drift occurrences.

The adoption of established Machine Learning (ML) models has also been extensively explored by authors. For instance, Xuan et al. (2020) [22] applied Bayesian theory to design a nonparametric, label-independent method based on the Polya-tree hypothesis test. The core concept involves decomposing the underlying data distribution into a multi-resolution representation, which simplifies the hypothesis testing process into recursive binomial tests.

In this sense, Kingetsu et al. (2021) [23] developed a label-independent proposal using Artificial Neural Networks (ANN). In summary, they created an inspector model based on ANN that emulates the decision boundary of the deployed model. This inspector model computes the distance between the decision boundary of the deployed model and the data points, thereby detecting variations in this distance to identify concept drift. Fuccellaro et al.(2022) [24] introduced a partially label-independent method called Partially Supervised Drift Detection (PSDD), which builds a Decision Tree with pure leaves that monitors changes within each leaf distribution to identify drift.

Gözüaçık and Can (2021) [25] presented the One-Class Drift Detector (OCDD), which monitors concept drift by using a one-class classifier applied over a sliding time window. This classifier is trained to decide whether new samples deviate from the established data distribution. The detector identifies drift based on the proportion of outliers found within the sliding window. This detector is partially label-independent, as it requires initializing the one-class classifier.

Exploring a different direction, some detectors have been specifically designed to identify drift in data streams with delayed labels. Zheng et al. (2019) [26] proposed a label-dependent efficient method for monitoring changes in the space of Shapley values. This method captures the individual contributions of features to the predictions made by a classifier. By tracking changes in each Shapley feature individually, the approach demonstrates its effectiveness, as each Shapley feature encapsulates the interactions among the original features in relation to the classifier's output.

In [27], a novel partially label-dependent ensemble method is introduced for detecting concept drift and covariate shift. This ensemble comprises six drift detection modules, each capturing distinct characteristics of incoming data, such as delayed classification error rates, and model uncertainty. It employs a customized majority voting strategy, reducing reliance on any single signal and enhancing robustness against anomalous data.

Finally, Melo et al. (2023) [28] introduced a partially label-dependent novel model monitoring tool based on Meta-Learning concepts. This algorithm is designed for data streams that exhibit concept drift and significant delays in target arrival. In summary, this detector leverages Meta Learning to identify patterns in existing changes within the data. The authors introduce a novel set of Meta Features derived from unsupervised concept drift metrics to enhance the effectiveness of the proposed approach.

3.2.3. Output classification

Lindstrom et al. (2013) [29] presented the Confidence Distribution Batch Detection (CDBD) approach for detecting concept drift, which explicitly identifies changes without requiring information about the true classes of test instances (label-independent). CDBD evaluates the distribution of classifier output confidences within a batch of test examples and compares it to a reference distribution derived from the training data. This comparative analysis provides a quantifiable measure of concept drift. When this measure exceeds a predefined threshold, concept drift is recognized, prompting an update of the classifier.

The detection of concept drift in partially labeled data streams has also been explored through window-based methodologies, which focus on the outputs of classifiers. Kim and Park (2016) [30] proposed method detects concept drift in unlabeled data streams by leveraging class-label information predicted by a classifier trained on a limited set of labeled data samples. The concept drift detection approach utilizes two distinct windows: reference and detection windows. The study introduces a drift detection mechanism that includes fixed reference windows, dynamic reference windows, and an ensemble of reference windows.

The research conducted by Sethi and Kantardzic (2017) [31] introduced an innovative label-independent drift detection method named Margin Density Drift Detection (MD3). This approach focuses on monitoring the quantity of samples within the uncertainty region of a classifier as a pivotal metric for drift detection. MD3 carefully observes the sample count in the classifier's uncertainty region, often referred to as its margin, to identify drift occurrences. After training, robust classifiers such as Support Vector Machines (SVM) or feature-bagged ensembles exhibit distinct regions of uncertainty known as margins. Since the computation of margin density relies solely on unlabeled data, it serves as a viable alternative to explicit labeled drift detection techniques for effectively tracking changes in $P(Y|X)$.

Heyden et al. (2024) [32] introduced the label-independent ABCD method, a change detection approach for high-dimensional data based on a Artificial Neural Network. It monitors an encoder-decoder model's loss (MSE) using an adaptive window size combined with statistical testing. The adaptive window allows ABCD to rapidly detect abrupt changes while identifying more subtle shifts over extended periods-changes that typically require a larger window. ABCD employs a statistical test based on Bernstein's inequality to enhance accuracy.

Within the domain of Artificial Neural Networks, Yu et al. [33] introduced Meta-ADD, and Wang et al. [34] proposed QuadCDD. Meta-ADD detects concept drift through a two-stage process: offline pre-training followed by online fine-tuning. In the pre-training phase, it operates on data streams with known drifts, extracting meta-features from error rates associated with different drift types. These meta-features are used to train a meta-detector via a prototypical neural network, with each drift type represented as a distinct prototype. During online detection, the meta-detector adapts to the target data stream using a lightweight, stream-based active learning strategy, enhancing real-time detection accuracy.

In contrast, QuadCDD (Quadruple-based Approach for Understanding Concept Drift in Data Streams) not only detects and predicts the onset of concept drift but also provides a detailed characterization via a quadruple representation: drift start, drift end, drift severity, and drift type. During pre-training, features capturing dataset characteristics under different drift types are extracted, and an initial model is trained to establish a performance baseline. The model is then fine-tuned on multi-domain data, improving adaptability and generalization. In the detection and decision-making phases, the quadruple representation serves as a structured input to guide accurate drift characterization and response.

In [35] and [36], the authors presented label-independent drift detectors specifically designed for Delayed Partially Labeled data streams. These detectors use a student-teacher (ST) learning paradigm, where the teacher model serves as the primary model and a secondary model, the student, is also constructed. The mimicking loss, a key component of this approach, is defined by the discrepancy between the predictions of the teacher and student for the same instance. Essentially, the loss of the student model acts as a proxy for the behavior of the primary model.

Finally, Gulcan and Can (2023) [37] introduced a label-independent concept drift detector specifically designed for multi-label data streams. This method capitalizes on the inherent label dependencies within the data by leveraging a label influence ranking technique. By utilizing a data fusion algorithm, it identifies dynamic temporal dependencies between labels and subsequently employs the generated ranking to detect concept drift

3.3. Summary of concept drift detectors

Table 3 presents a comparative summary of all discussed concept drift detectors designed for delay and partially labeled data streams, which are the main focus of our investigation. Our analyses are based on the following characteristics: Architecture (A) categorizes detectors based on our taxonomy; Model Dependency (MD) indicates whether detectors rely on machine learning models; Machine Learning Problem (MLP) specifies the type of learning problem (Classification or Regression); Label Availability (LA) informs whether the data is unlabeled or partially labeled; Label Delay (LD) presents the type of delay interval; and Drift Type (DT) highlights the types of classical concept drifts considered in the experiments.

Table 3
Summary of Concept Drift detectors for delay and partially labeled data streams.

Algorithm	(A)	(MD)	(MLP)	(LA)	(LD)	(DT)
CDBD [29]	OC	Yes	C	UNL	VA	Und
FRW, MRW, ERW [30]	OC	Yes	C	PA	IM	Und
IKS [20]	DM	No	C	UNL	VA	Und
MD3 [31]	OC	Yes	C	PA	VA	Und
MDE [16]	DM	No	C	UNL	VA	A, G
L-CODE [26]	DM	Yes	C	PA	IM	Und
D3 [21]	DM	Yes	C	UNL	VA	Und
IBDD [14]	RDM	No	C	UNL	VA	A, I, O
BNDM [22]	DM	Yes	C	UNL	VA	Und
BADB [23]	DM	Yes	C	UNL	VA	A, G
OCDD [25]	DM	Yes	C	UNL	VA	Und
CDCSDE [27]	DM	Yes	C	PA	FI	A, G
PSDD [24]	DM	Yes	C	PA	IM	A
STUDD [36]	OC	Yes	C	UNL	VA	A, I
CDDD [15]	RDM	No	C	UNL	VA	I, A, G, R
LD3 [37]	OC	Yes	C	UNL	VA	I, R, A
MtL [28]	DM	Yes	C, R	PA	VA	Und
ABCD [32]	OC	Yes	C	UNL	VA	Und
Meta-ADD [33]	ER	Yes	C	PA	VA	A, G, I
QuadCDD [34]	OC	Yes	C	PA	VA	A, G, I

Legend – **Und:** Undefined; **(A) Architecture:** DM: Distribution Monitoring, OC: Output Classification; RDM: Raw Data Monitoring; ER: Error Rate **(MD) Model Dependency; (MLP) Machine Learning Problem:** C: Classification, R: Regression; **(LA): Label Availability:** UNL: Unlabelled, PA: Partially **(LD) Label Delay:** VA: Variable, IM: Immediate, FI: Fixed **(DT) Drift Type:** A: Abrupt, I: Incremental, O: Oscillating, G: Gradual, R: Reoccurring.

Table 3 shows that the majority of methods adopt Distribution Monitoring (DM) or Output Classification (OC) architectures, whereas relatively few employ Raw Data Monitoring (RDM) or error-rate-based detection (ER). Most approaches are designed for classification tasks, with only MtL addressing both classification and regression. While many algorithms operate under unlabelled data with variable label delays, partially labeled streams or fixed/immediate delays are less commonly considered. Information on drift type is often unspecified; among the reported cases, abrupt (A) and gradual (G) drifts are most prevalent, whereas incremental (I), oscillating (O), and recurring (R) drifts are addressed by fewer methods.

4. Evaluation methods

Traditional evaluation metrics and methods (e.g., accuracy, precision, recall, F-measure, and area under the ROC curve) represent an essential aspect in assessing the performance of general learning models. However, they were not explicitly tailored to handle scenarios involving delayed and partially labeled data streams, which present inherent complexities, as discussed by Gomes et al. (2022) [2]. The first challenge lies in ensuring the timely availability of labels for model training. Additionally, since the data is collected as an open-ended sequence, identifying the optimal timing to assess models can also influence the accurate evaluation of their performance. In the context of Concept Drift, there are methods specifically designed to assess the performances of detectors, as outlined in [38–41]. However, as analyzed in this section, they also present limitations when handling delayed and partially labeled data.

Our evaluation analyses were conducted from two main perspectives: (i) learner-dependent - focusing on evaluation processes centered on the performance of the learners used by concept drift detectors; and (ii) learner-independent - involving studies that assess the concept drift detectors without directly analyzing the performance of their learners. Aiming to systematically structure our discussion, we organized the aforementioned concept drift detectors into four dimensions. Firstly, we use "Learner Algorithm" (LA) to specify the base learner employed in formulating or testing the authors' proposals. Secondly, we use "Learner Evaluation" (LE) to highlight the metrics/methods considered to assess the performance of the learning models. Next, "Drift Detector Evaluation" (DDE) shows the metrics/methods employed for evaluating concept drift detectors. Finally, "Data Stream" (DS) indicates whether the data utilized in the research experiments was real or synthetic.

4.1. Evaluation of learner-dependent detectors

One of the earliest evaluation proposals is referred to as test-then-train, also known as the prequential approach [38]. In this approach, an unlabeled instance is sent to a pre-trained model, and the predicted label is then compared with the actual one to determine whether the model needs to be updated.

Another evaluation method commonly considered is called cross-validated prequential approach [41], in which models are concurrently trained and tested on data folds to estimate their predictive performance more accurately. Bifet et al. (2015) [41] proposed three distinct strategies for cross-validated prequential evaluation: k-fold distributed cross-validation, k-fold distributed split-validation, and k-fold distributed bootstrap validation. These strategies share the common feature of training and testing k models in parallel, though they differ in how the folds are constructed.

However, these approaches become impractical when dealing with delayed labels, as the time gap between the availability of the unlabeled instances and their corresponding true labels can be significant [39]. A potential solution to this issue is the implementation of a buffer to temporarily store data with delayed labels. Once the expected label becomes available, the data can be retrieved from the buffer and evaluated by the model, thus comparing predicted and expected labels. The main problem with this approach is the significant delay in obtaining the actual label. In dynamic systems, the model may evolve, thus making the initial prediction less relevant to the current state of the system. Aiming to overcome this situation, Grzenda et al. (2020a) [39] introduced a re-evaluation technique, in which new predictions are continuously performed and evaluated with labels available over time until the arrival of the true label of the instances that the label is delayed. The main advantage of this technique is the possibility of using models that are learning from data collected while the true label is still unavailable. Although this technique is important for handling delayed data streams, it does not tackle the challenge of evaluating concept drift detectors. An advancement over prior research is presented by Grzenda et al. (2020b)[40], where the test-then-train performance measures are enhanced by introducing Intermediate Performance Measures (IPMs) that provide quantitative evaluations for intermediate predictions.

4.2. Evaluating learner-independent drift detectors

The studies published in [35,36] are among the pioneering studies, evaluating concept drift detectors on delay and partial labeling data streams. The evaluation proposed by the authors is based on the key metrics Mean Time between False Alarms (MTFA), Mean Time to Detection (MTD), False Alarm Ratio (FAR), and Missed Detection Ratio (MDR). In a nutshell, MTFA means the average time elapsed between false alarms triggered by the detector under evaluation. MTD quantifies the average amount of data required by the detector to detect changes. FAR computes the ratio of data in which the detector triggers at least one false alarm. Finally, MDR represents the proportion of data where the detector fails to detect changes. Later, as new detectors emerged, additional metrics were introduced or adapted from traditional machine learning, such as "True Positive" used as "Drift Detected".

In Table 4, we explore how authors use the evaluation process to assess new detectors, focusing on four key dimensions: Learner Algorithm (LA), Learner Evaluation (LE), Drift Detection Evaluation (DDE), and Data Stream (DS). The listed approaches encompass a diverse range of learning models, from classical algorithms such as Decision Trees (DT), Support Vector Machines (SVM), and Naive Bayes (NB) to more advanced techniques like Convolutional Neural Networks (CNN), Gradient Boosting (XGB, LightGBM), and ensemble strategies. Evaluation metrics for learners include accuracy-oriented measures (Accuracy, F1, Cohen's Kappa) as well as precision, recall, and error-based indicators (MSE, Gmean, Specificity, Wilcoxon pair ranksum). Drift detector evaluations focus on detection reliability, delay, false alarm rates, and missed detections (e.g., TPD, ADD, FAR, MTD). The data streams used encompass both synthetic (S) and real-world (R) scenarios, capturing the range of experimental contexts considered in the analyzed studies.

The IKS, IBDD, MtL, L-CODE, CDDD, OCDD, and D3 detectors were not analyzed by DDEs; the authors only assessed the performance of the learner algorithms (LA), in which accuracy was the main considered metric. However, as discussed by Bifet (2017) [42], metrics focused on the learner might hide detectors' inefficiency, especially if high accuracy is due to frequent retraining, thus leading to unnecessary model retraining and wasting resources without addressing real concept drifts. On the other hand, there are studies that explicitly assessed the efficacy of detectors, as shown in Column DDE. An important observation from those studies is the adoption of synthetic data streams due to the need for ground truth information, missing in most real-world streams [43].

From the perspective of CD detectors in delayed and partially labeled data streams, the most considered metrics include True Positive Drifts Detected (TPD), True Negative Drifts Detected (TND), False Positive Drifts Detected (FPD), False Negative Drifts Detected (FND), Total Drifts Detected (TDD), and Drift Alarm Error Average (DAA). Specifically, TPD, TND, FPD, and FND values can be utilized to construct a confusion matrix, enabling the extraction of specific metrics tailored to the CD detector. Additionally, TDD and DAA metrics provide a broader perspective on the detector's overall performance. In scenarios where label availability is delayed, it is also recommended to use Drift Delay Detection Average (DDA) and True Detection Delay Average (TDDA). These metrics provide valuable insights into the delay in detecting drift, illustrating the discrepancy between when the detector signals concept drift and the true occurrence of the drift.

5. Experimental analyses

The experiments presented in this section were designed to assess the performance of concept drift methods when confronted with delayed and partially labeled data streams. The first experiment specifically addresses the classification scenario described in Section 2, focusing on Item (iv), where the data is partially labeled, i.e., a subset of the labels remains permanently unavailable. In this experiment, we varied the proportion of missing labels, specifically testing scenarios with 95%, 96%, 97%, 98%, and 99% of the labels unavailable. For each percentage of unlabeled data, we performed 30 repetitions of the experiment to avoid results by chance. The second experiment, corresponding to Item (ii) of Section 2, investigates scenarios where label availability is delayed for both training and concept drift detection. We evaluated delay sizes of 1,000, 2,000, 4,000, 8,000, and 10,000, performing 30 repetitions for each configuration. To maintain a concise discussion, this section focuses on the results obtained under the most challenging scenario: 99% unlabeled data with a delay of 10,000 instances. The complete set of results for all other configurations is provided in the appendices.

The experimental setup followed the prequential evaluation method, with modifications to either exclude or delay a specified percentage of labels. The data streams analyzed in our experiments were produced by

Table 4
Evaluation of Concept Drift detectors for delayed and partially labeled data streams.

Algorithm	(LA)	(LE)	(DDE)	(DS)
PSDD	DT	Und	FPD, FND, DD	S, R
STUDD	RF, XGB, DT, RC, LR, NB, SVM	CK	MDR, FAR, MTD, MTFA	S, R
IKS	NN, NB, DT	A	Und	S, R
IBDD	RF	A	Und	R
BADB	ANN	A	FPD, FND, TPD, DDy	S, R
MDE	Und	Und	ARL, TTD, NFA, MDR	S, R
BNDM	SVM	A	DR	S, R
LD3	CC, NB	A	ADD, TPD, FPD	S, R
ABCD	Encoder-Decoder Model	Und	P, R, F1, MTD	S, R
MtL	RF, SVM, DT, LR, LG	F1, R, P, CK, MSE	Und	R
L-CODE	RF	A	Und	S, R
CDCSDE	CNN	Und	MTFA, MTD, MDR, DD,MA	S, R
FRW, MRW, ERW	SVM, LDA	A, F1	P, R, F1	S, R
MD3	SVM, RS	A	DD, FPD	S, R
CDDD	MLP, NB	A, F1, G, R, S, W	Und	S, R
OCDD	HT, OSVM	A	Und	S, R
CDBD	SVM	A	TPD, FPD, FND, TND, P, R	S
D3	LR, HT	A	Und	S, R
Meta-ADD	FCN, RNN	Und	MA, F1	S, R
QuadCDD	LSTM	A, F1	DDy	S, R

Legend – **Und:** Undefined; **(LA) Learner Algorithm:** FCN: Fully Connected Neural Network, RNN: Recurrent Neural Network, RF: Random forest, SVM: Support vector machine, DT: Decision tree, LR: Logistic regression, RC: Ridge classifier, NB: Naive Bayes, XGB: Extreme Gradient Boosting, NN: Nearest Neighbor, RS: Random Subspace, (LDA) Linear Discriminant Analysis, CNN: Convolution Neural Network, LG: LightGBM, DW: DWMIL, LEA: Learn++, CC: Classifer Chain, MLP: MultiLayer Perceptron Classifier, ANN: Artificial Neural Network, HT: Hoeffding Tree, OSVM: One-class SVM; **(LE) Learner Evaluation:** CK: Cohen's kappa, A: Accuracy, F1: F1-measure, R: Recall, P: Precision, MSE: Mean Squared Error, G: Gmean, S: Specificity, W: Wilcoxon pair ranksum; **(DDE) Drift Detector Evaluation:** TPD: True Positive Detection, TND: True Negative Detection, FPD: False Positive Detection, FND: False Negative Detection, DDy: Detection Delay, ADD: Average Detection Delay, MDR: Missed Detection Ratio, FAR: False alarm ratio, MTD: Mean Time to Detection, MTFA: Mean time between false alarms, MA: Mean accuracy, DD: Drifts Detected, P: Precision, R: Recall, F1: F1-measure, DR: Drift-Rates, ARL: Average Running Length, TTD: Time To Detection, NFA: Percentage did not false alarm; **(DS) Data Stream:** S: Synthetic, R: Real.

the MOA framework,[1] using the following generators: Agrawal, LED, Mixed, Random RBF, Sine, and Waveform. The Agrawal generator was utilized twice: Agrawal1 employs its first five functions (F1 to F5), while Agrawal2 utilizes its remaining ones (F6 to F10). In our experiments, the data streams consisted of 100,000 instances with abrupt and gradual concept drifts starting at the following time instants: 20,000, 40,000, 60,000, and 80,000. Concept drifts were simulated by alternating between different concepts. In all gradual drift datasets, each drift spanned 500 instances and was generated using a probability function: starting 250 instances before the change point and continuing until 250 instances after, class labels gradually transitioned from representing the old concept to the new one. At the exact change point, the probability of each concept was equal (50%). In contrast, abrupt drifts were defined with a transition window of a single instance, meaning the shift occurred instantaneously. In this study, abrupt drifts were introduced at four predefined points in the data stream (20k, 40k, 60k, and 80k). Finally, we also used the Hoeffding Tree algorithm for model-dependent drift detectors in the classification problems [44].

Furthermore, the experiments followed the widely adopted methodology of resetting base models upon the detection of changes and considered the following drift detectors: DDM, EDDM, ADWIN, ECDD, STEPD, SeqDrift2, SEED, $HDDM_A$, $HDDM_W$, RDDM, Two-sample Kolmogorov-Smirnov test (Test_Ks), Wilcoxon rank-sum test (Test_Wrs), Two-sample *t*-test (Test_Tt), Multivariate ensemble drift detection, and Studd. For RDDM, we use two versions with different set of parameters. The configuration for $RDDM_{30}$ uses the default DDM parameters with $n = 30$, $\alpha_w = 2$, and $\alpha_d = 3$. For RDDM, we considered its own proposed default parameters: $n = 129$, $\alpha_w = 1.773$, and $\alpha_d = 2.258$. In Table 5, we summarized all parameters used by the analyzed algorithms in our experiments.

The first results, listed in Table 6, summarize results obtained from running those detectors on partially labeled (99%) data streams with abrupt changes, evaluated using TPD, TDD, and TDDA. It is important to emphasize that the large number of experiments, combined with the complexity of the data stream configurations, led to high prediction errors for some algorithms. As a result, the error distributions may deviate from normality, potentially compromising analyses based on means and standard deviations. To address this issue, we chose to summarize the results using the median and the interquartile range (IQR), a more robust measure of variability, in our scenarios, that captures the spread of the middle 50% of the data. The IQR is calculated as the difference between the third quartile (Q3: 75%) and the first quartile (Q1: 25%).

According to the TPD results, one may notice the Wilcoxon ranksum (Test_Wrs) test presents the best performance. However, a closer examination using TDD reveals that Test_Wrs detects over 1000 concept drifts in certain data streams, despite there being only four actual drifts. This large number of false positives inflated the TPD result, as it included the four genuine drifts.

Regarding the TDDA metric, SeqDrift2 showed the most significant delay in detecting true drifts across the majority of data streams. In contrast, Test_Wrs exhibited the shortest delay. However, as previously discussed, this result is due to its high rate of false drift detections. Notably, algorithms like Multivariate and Studd failed to detect any drifts in several data streams. Upon analysis of Table A.1, a pattern similar to the previous experiment for gradual concept drift emerged. Test_Wrs achieved the highest performance in terms of TPD but with a notable increase in TDD. Consequently, Test_Wrs yielded a lower TDDA. Conversely, SeqDrift2 exhibited the highest TDDA across the majority of data streams.

[1] https://moa.cms.waikato.ac.nz/details/classification/streams/

Table 5
Parameters used by every Concept Drift method.

Algorithm	Parameters
DDM	MinNumInstances = 30; WarningLevel = 2; OutControlLevel = 3
EDDM	—
ADWIN	$\delta = 0.002$
ECDD	MinNumInstances = 30; $\lambda = 0.2$
STEPD	windowSize = 30; $\alpha Drift = 0.003$; $\alpha Warning = 0.05$
SeqDrift2	$\delta = 0.01$; blockSeqDrift2 = 200
SEED	$\delta = 0.05$; blockSize = 32; $\epsilon = 0.01$; $\alpha = 0.8$; compressTerm = 75
Multivariate	driftDetectionMethod = PageHinkley; agreement = 20%
HDDM_A	driftConfidence = 0.001; warningConfidence = 0.005; oneSidedTest = Two-sided
HDDM_W	driftConfidence = 0.001; warningConfidence = 0.005; $\lambda = 0.050$; oneSidedTest = One-sided
RDDM_30	$\alpha_w = 2$; $\alpha_d = 3$
RDDM	$n = 129$; $\alpha_w = 1.773$; $\alpha_d = 2.258$
Test_Ks	windowSize = 10; threshold = 0.05
Testz_Wrs	windowSize = 10; threshold = 0.05
Testz_Tt	windowSize = 10; threshold = 0.05
Sudd	baseLearner = AdaptiveRandomForest; studentLearner = AdaptiveRandomForest; driftDetectionMethod = PageHinkley; sizeBatchTrain = 500;

Table 6
Metrics: TPD, TDD, and TDDA. Partially labeled (99%) with abrupt CD. Data Stream Size 100k.

Drift Type and Metric	Data Stream	DDM HDDM$_A$	EDDM HDDM$_W$	ADWIN RDDM$_{30}$	ECDD RDDM	STEPD Test_Ks	SeqDrift2 Test_Wrs	SEED Test_Tt	Multivariate Studd
	AGRAW$_1$	1.00 (1.00)	4.00 (1.75)	1.00 (1.00)	2.00 (1.00)	1.00 (1.00)	1.00 (0.00)	1.00 (0.00)	1.00 (1.00)
		1.00 (1.00)	2.00 (0.00)	1.00 (0.00)	2.00 (1.00)	2.00 (2.00)	**4.00 (0.00)**	2.00 (2.00)	0.00 (0.00)
	AGRAW$_2$	0.00 (2.00)	3.00 (2.00)	0.00 (0.00)	3.00 (1.00)	3.00 (1.00)	1.00 (1.00)	2.00 (1.00)	1.00 (1.00)
		1.00 (2.00)	2.00 (0.75)	0.00 (1.00)	2.00 (3.00)	2.00 (0.75)	**4.00 (0.00)**	2.00 (1.00)	0.00 (0.00)
	LED	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (1.00)	1.00 (1.00)	2.00 (1.00)	3.00 (1.00)	0.00 (0.00)
		0.00 (0.00)	0.00 (0.00)	1.00 (0.00)	0.00 (0.00)	2.00 (1.00)	**4.00 (0.00)**	2.00 (2.00)	0.00 (0.00)
ABRUPT TPD	**MIXED**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	3.00 (1.00)	4.00 (0.75)	0.00 (0.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	3.00 (1.00)	**4.00 (0.00)**	3.50 (1.00)	0.00 (0.00)
	RandRBF	**4.00 (0.00)**	**4.00 (0.00)**	1.00 (1.00)	**4.00 (0.00)**	4.00 (1.00)	2.00 (0.00)	2.00 (1.00)	0.00 (0.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	3.00 (2.00)	**4.00 (0.00)**	3.00 (1.75)	0.00 (0.00)
	SINE	3.50 (1.00)	4.00 (1.00)	4.00 (1.00)	4.00 (1.00)	4.00 (1.00)	2.00 (1.00)	4.00 (1.00)	0.00 (0.00)
		4.00 (1.00)	4.00 (1.00)	3.00 (1.00)	4.00 (1.00)	3.00 (1.00)	**4.00 (0.00)**	3.00 (1.75)	0.00 (0.00)
	WAVEE	0.00 (0.00)	1.00 (2.00)	0.00 (0.00)	1.00 (1.75)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
		0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	2.00 (2.00)	**4.00 (0.00)**	2.00 (1.00)	0.00 (0.00)
	AGRAW1	148.00 (155.25)	700.50 (490.25)	**9.50 (101.25)**	212.00 (168.50)	93.50 (189.00)	88.50 (175.50)	88.00 (152.00)	220.50 (348.75)
		102.00 (115.25)	186.50 (186.00)	79.50 (129.25)	123.00 (202.50)	252.50 (277.00)	1760.00 (472.50)	295.50 (447.50)	0.00 (0.00)
	AGRAW2	0.00 (139.75)	385.50 (389.50)	0.00 (0.00)	373.50 (253.50)	274.00 (356.00)	**6.00 (135.25)**	194.00 (233.75)	220.50 (348.75)
		30.00 (139.25)	200.00 (190.00)	0.00 (90.50)	107.00 (271.00)	396.00 (284.25)	3099.50 (651.25)	391.50 (480.75)	0.00 (0.00)
	LED	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (34.25)	**224.50 (302.75)**	283.00 (258.00)	410.00 (341.00)	0.00 (0.00)
		0.00 (0.00)	0.00 (0.00)	80.00 (120.50)	0.00 (0.00)	352.50 (326.50)	2252.00 (449.50)	420.00 (317.75)	0.00 (0.00)
ABRUPT TDD	**MIXED**	378.50 (139.75)	423.50 (188.25)	348.00 (316.75)	416.00 (331.25)	396.00 (261.00)	**294.50 (259.00)**	665.50 (363.00)	0.00 (0.00)
		364.50 (130.75)	368.50 (194.00)	376.50 (276.50)	409.00 (270.25)	388.00 (410.75)	2492.00 (490.00)	406.50 (403.75)	0.00 (0.00)
	RandRBF	377.00 (216.50)	458.50 (323.75)	**94.50 (133.00)**	359.50 (270.00)	706.50 (365.75)	177.00 (282.00)	421.00 (315.75)	0.00 (0.00)
		372.50 (234.00)	357.50 (247.25)	372.50 (331.75)	418.50 (228.50)	363.50 (375.50)	2486.00 (441.75)	389.00 (356.00)	0.00 (0.00)
	SINE	281.50 (148.50)	404.50 (348.50)	**269.50 (136.75)**	455.00 (367.25)	336.50 (178.50)	280.00 (265.75)	574.50 (357.25)	0.00 (0.00)
		292.00 (191.25)	314.50 (243.50)	309.50 (301.25)	334.50 (278.50)	346.50 (317.00)	2716.00 (554.75)	336.50 (366.00)	0.00 (0.00)
	WAVEE	0.00 (0.00)	160.00 (362.75)	0.00 (0.00)	**74.00 (191.50)**	0.00 (36.50)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
		0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	367.50 (234.50)	2563.50 (1027.25)	295.00 (359.75)	0.00 (0.00)
	AGRAW1	48.99 (57.38)	16.17 (13.65)	49.16 (156.20)	44.31 (51.40)	40.76 (74.36)	124.44 (320.85)	118.56 (162.52)	**7.62 (44.41)**
		61.61 (100.37)	43.06 (44.19)	56.59 (203.29)	51.56 (50.84)	52.78 (63.30)	8.26 (3.54)	36.23 (67.38)	0.00 (0.00)
	AGRAW2	0.00 (8.31)	30.02 (19.08)	0.00 (0.00)	11.65 (16.25)	37.95 (101.80)	17.82 (81.25)	93.05 (148.20)	7.62 (44.41)
		17.41 (71.68)	33.45 (58.53)	0.00 (32.45)	29.16 (68.36)	26.36 (35.53)	**3.81 (2.37)**	40.64 (64.78)	0.00 (0.00)
	LED	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (133.08)	29.31 (44.96)	71.69 (160.07)	68.37 (35.66)	0.00 (0.00)
		0.00 (0.00)	0.00 (0.00)	91.89 (101.77)	0.00 (0.00)	51.44 (62.35)	**6.03 (3.93)**	42.16 (29.48)	0.00 (0.00)
ABRUPT TDDA	**MIXED**	13.70 (6.26)	8.88 (4.60)	46.28 (55.29)	**4.16 (3.32)**	11.74 (8.13)	63.48 (235.41)	21.37 (12.23)	0.00 (0.00)
		8.80 (5.33)	13.24 (7.40)	15.97 (12.20)	8.07 (7.05)	9.88 (13.43)	5.60 (3.87)	9.92 (14.72)	0.00 (0.00)
	RandRBF	32.76 (22.77)	17.38 (11.05)	116.85 (158.70)	8.30 (5.88)	28.03 (23.78)	80.01 (215.62)	43.63 (42.69)	0.00 (0.00)
		12.69 (7.47)	19.88 (16.01)	37.70 (37.71)	18.05 (12.98)	13.01 (29.40)	**5.28 (2.84)**	12.54 (30.24)	0.00 (0.00)
	SINE	18.12 (12.16)	13.54 (13.36)	78.62 (37.17)	**3.89 (5.77)**	14.71 (11.27)	81.30 (196.34)	28.62 (20.04)	0.00 (0.00)
		14.29 (9.52)	16.30 (12.07)	21.36 (25.59)	10.33 (9.41)	12.02 (23.47)	4.55 (3.06)	28.71 (45.73)	0.00 (0.00)
	WAVEE	0.00 (0.00)	11.69 (53.41)	0.00 (0.00)	65.17 (225.78)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
		0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	47.15 (52.46)	**3.85 (3.28)**	33.14 (47.64)	0.00 (0.00)

In Table 7, we repeated the previous experiments, but this time changing the label behavior. Instead of missing labels, we introduced delays of 10,000 instances. The figure shows that most detectors, except for Multivariate and Studd, successfully identified the true concept drifts, as indicated by the TPD metric. When analyzing the TDD plots, it becomes apparent that some detectors triggered numerous detections beyond the four expected drifts, with Test_Wrs detecting over 1,000. Regarding TDDA, Test_Wrs exhibited the shortest delay, likely due to its high number of false positives.

Analyzing these results reveals that delayed and partially labeled data streams significantly impact the performance of detectors in general. The DDM algorithm exhibited limited effectiveness in detecting

Table 7
Metrics: TPD, TDD, and TDDA. Maximum Delay Labels 10.000 with abrupt CD. Data Stream Size 100k.

Drift Type and Metric	Data Stream	DDM $HDDM_A$	EDDM $HDDM_W$	ADWIN $RDDM_{30}$	ECDD RDDM	STEPD Test_Ks	SeqDrift2 Test_Wrs	SEED Test_Tt	Multivariate Studd
	AGRAW$_1$	**4.00 (0.00)**	3.00 (1.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	3.00 (2.75)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
	AGRAW$_2$	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	3.00 (2.75)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
	LED	1.50 (1.00)	0.00 (0.00)	2.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	2.00 (0.75)	**4.00 (0.00)**	0.00 (0.00)
		2.00 (0.00)	**4.00 (0.00)**	2.00 (0.00)	2.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	1.00 (0.00)
ABRUPT TPD	**MIXED**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	2.00 (0.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
	RandRBF	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	2.00 (1.00)
	SINE	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	1.00 (0.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
	WAVEE	0.00 (0.75)	0.00 (1.00)	0.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)	3.00 (1.00)	0.00 (0.00)
		0.00 (1.00)	**4.00 (0.00)**	2.00 (2.00)	4.00 (2.75)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
	AGRAW1	4.00 (1.00)	18.50 (9.50)	11.00 (2.00)	131.00 (15.50)	32.00 (5.00)	23.50 (4.00)	65.00 (9.50)	5.00 (5.75)
		5.00 (1.00)	22.50 (4.00)	**4.00 (0.00)**	5.00 (2.00)	295.00 (24.25)	1559.00 (43.00)	260.50 (26.25)	0.00 (0.00)
	AGRAW2	16.50 (12.75)	16.50 (5.75)	6.00 (1.75)	159.00 (13.50)	47.00 (23.25)	12.50 (3.00)	43.00 (6.50)	5.00 (5.75)
		4.00 (0.75)	9.00 (6.00)	**4.00 (0.00)**	4.00 (0.75)	142.50 (13.75)	2576.50 (23.25)	131.50 (11.50)	0.00 (0.00)
	LED	2.00 (1.75)	0.00 (0.00)	**5.00 (0.75)**	69.00 (11.50)	84.50 (46.25)	10037.00 (7505.00)	411.50 (58.75)	0.00 (0.00)
ABRUPT TDD		2.00 (1.00)	23.00 (8.75)	2.00 (0.00)	2.00 (0.00)	297.00 (47.75)	2089.50 (85.75)	286.50 (17.75)	1.00 (0.00)
	MIXED	4.00 (4.00)	20.00 (9.75)	**4.00 (0.00)**	119.50 (13.25)	34.00 (8.75)	8.00 (0.00)	38.00 (10.75)	3.00 (2.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	9.00 (10.00)	70.00 (19.75)	2447.00 (77.00)	55.50 (23.50)	0.00 (0.00)
	RandRBF	4.00 (1.00)	9.00 (2.00)	8.00 (2.00)	65.50 (6.00)	47.00 (13.00)	10108.50 (73.75)	519.00 (28.00)	0.00 (0.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	4.00 (1.00)	141.00 (17.00)	2381.00 (43.50)	123.00 (21.00)	2.00 (2.00)
	SINE	7.50 (3.00)	36.00 (24.00)	**4.00 (0.00)**	152.50 (19.00)	23.50 (12.25)	8.00 (0.00)	62.00 (9.00)	1.00 (1.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	6.00 (5.75)	83.00 (15.00)	2570.50 (52.50)	81.50 (25.25)	0.00 (0.00)
	WAVEE	0.00 (1.00)	**4.00 (9.50)**	0.00 (1.75)	153.00 (19.50)	26.50 (13.00)	0.00 (0.75)	8.00 (5.00)	0.00 (0.00)
		1.00 (1.75)	10.00 (4.00)	3.50 (4.75)	8.00 (6.00)	230.00 (43.00)	2473.00 (58.25)	276.50 (47.75)	0.00 (0.00)
	AGRAW1	10301.75 (2124.03)	1952.90 (1292.06)	3659.64 (674.67)	34.88 (61.80)	491.65 (370.52)	1268.59 (549.94)	108.38 (80.23)	3071.06 (6125.82)
		7952.30 (2247.32)	1354.24 (243.98)	10349.75 (67.44)	8178.20 (3407.35)	**4.05 (4.75)**	0.10 (0.10)	6.37 (8.05)	0.00 (0.00)
	AGRAW2	2473.92 (2284.28)	2285.48 (965.88)	6701.34 (2050.28)	11.40 (11.40)	500.20 (335.14)	2749.45 (939.95)	393.17 (169.57)	3212.27 (6125.82)
		9911.75 (1657.44)	3425.00 (5136.50)	10191.88 (15.25)	10110.88 (1521.98)	83.82 (14.52)	**0.04 (0.03)**	88.67 (11.56)	0.00 (0.00)
	LED	4138.09 (2982.25)	0.00 (0.00)	1401.60 (188.00)	49.91 (45.49)	125.61 (200.22)	2.68 (303.64)	2.54 (3.28)	0.00 (0.00)
ABRUPT TDDA		2467.25 (1445.12)	308.61 (518.81)	8288.50 (508.62)	3721.75 (130.38)	2.23 (3.65)	**0.04 (0.05)**	4.09 (3.33)	11294.00 (1482.00)
	MIXED	10074.25 (5039.60)	1973.54 (750.47)	10032.00 (0.00)	13.07 (14.75)	152.74 (175.26)	5100.00 (0.00)	420.51 (398.15)	5490.25 (3149.28)
		9947.62 (576.25)	10010.50 (2.00)	10079.62 (8.75)	4465.34 (8190.23)	68.11 (90.50)	**0.04 (0.02)**	130.70 (156.60)	0.00 (0.00)
	RandRBF	10109.50 (1942.00)	4473.15 (984.38)	5016.00 (1273.90)	48.38 (22.34)	340.46 (250.79)	3.05 (1.74)	1.26 (0.75)	0.00 (0.00)
		9340.75 (1284.75)	10012.62 (2.06)	10110.50 (6.06)	10063.12 (2020.28)	29.11 (21.53)	**0.06 (0.05)**	54.71 (47.44)	10439.00 (2065.67)
	SINE	5402.74 (2241.14)	1063.84 (821.72)	10032.00 (0.00)	12.11 (8.00)	1060.20 (581.29)	5100.00 (0.00)	137.03 (151.15)	15175.50 (4457.62)
		9918.88 (155.25)	10012.50 (2.19)	10087.75 (7.69)	6701.08 (5917.10)	247.41 (55.19)	**0.04 (0.02)**	257.58 (67.01)	0.00 (0.00)
	WAVEE	0.00 (21.00)	0.00 (113.20)	0.00 (0.00)	10.36 (13.70)	348.44 (503.85)	0.00 (0.00)	2030.29 (2371.54)	0.00 (0.00)
		0.00 (3170.12)	2042.85 (1371.54)	2564.14 (3683.07)	2196.00 (2390.44)	4.73 (8.81)	**0.02 (0.03)**	3.59 (5.92)	0.00 (0.00)

concept drift in partially labeled data streams. While it correctly identified approximately four true drifts across the MIXED, RandRBF, and SINE streams, it also produced over 200 detected drifts in total, reflecting a high false alarm rate. Its performance improved under delayed-label scenarios, where the incidence of false alarms decreased. EDDM followed a detection pattern similar to DDM but under delayed-label conditions performed worse, generating substantially higher false alarm rates.

The Multivariate and Studd methods demonstrated minimal ability to detect both abrupt and gradual drifts under partially and delayed labeling conditions. Statistical test-based methods (Test_Ks, Test_Wrs, and Test_Tt) produced the highest false alarm rates. Although these methods reported a large number of drifts, their accuracy in correctly identifying true drifts in partially labeled streams remained low.

$HDDM_A$ exhibited substantial detection delays for both abrupt and gradual drifts in partially labeled and delayed-label scenarios. In contrast, $HDDM_W$ showed low detection delays but high false alarm rates under partially labeled conditions, while experiencing significant detection delays in delayed-label data streams.

ADWIN, ECDD, $RDDM_{30}$, and RDDM consistently exhibited both high false alarm rates and substantial detection delays across the evaluated scenarios. Similarly, STEPD, SeqDrift2, and SEED also showed elevated false alarm rates in partially labeled and delayed-label streams. These results underscore that designing effective concept drift detectors for such data streams remains a critical and open research problem, particularly given the increasing importance of temporal system modeling.

6. Conclusion and perspectives

In this paper, we explored concept drift detectors in delayed and partially labeled data streams, a setting that accurately reflects numerous real-world challenges faced in applying machine learning to data streams. We have covered different aspects of drift detectors within this context, including traditional and recent concept drift detector methods, their online adaptations, and the importance of conducting fair evaluations.

We propose a taxonomy to synthesize the current literature on concept drift detection in delayed and partially labeled data streams. This taxonomy provides a comprehensive overview of the state of the art, offering a systematic way to categorize and compare different drift detection methods. It also serves as a foundation for future research advancements, guiding researchers in identifying gaps and potential areas for improvement in the field.

In this context, we found that drift detection methods monitored raw data streams using different techniques. For instance, the IBDD detector [14] employs image-based methods, while the CDDD detector [15] relies on centroid distance analysis. Other detectors focus on distribution monitoring and utilize statistical approaches such as the Kolmogorov-Smirnov test [17–20] and the Polya-tree hypothesis test [22]. Additionally, ensemble methods [16], artificial neural networks [23], and sliding window algorithms [21,25] have been applied for distribution monitoring.

Beyond tracking data distribution, some detectors focus on monitoring the output classification. For example, MD3 [31] tracks the number of samples falling within a classifier's uncertainty region; ABCD [32] observes the mean squared error (MSE) of an encoder-decoder model using an adaptive window size paired with statistical testing; and STUDD [36] detects changes by comparing the predictions of a teacher-student model for the same input instance.

This paper emphasized a critical analysis of state-of-art evaluations when assessing drift detectors in delayed and partially labeled data streams. We provide a comprehensive discussion of evaluation procedures, offering researchers valuable guidance to design and assess their studies more effectively. In addition, we also evaluated the effectiveness of the concept drift detectors discussed in this manuscript. Through a series of carefully designed experiments, we analyzed their performance on data streams with partial and delayed labels. We conducted tests under both abrupt and gradual concept drift scenarios, using seven distinct types of synthetic data streams to ensure robust evaluation. These experiments provide a comprehensive and rigorous assessment of the detectors' performance.

We highlighted that traditional concept drift detectors, which rely on labeled data, tend to achieve high TPD rates. However, their high TDD values indicate that these detectors remain ineffective in distinguishing real drifts in delayed and partially labeled data streams. Similarly, recent methods designed to operate without labels also produced performance results comparable to those of traditional detectors, highlighting the persistent challenge of detecting drifts accurately in such complex scenarios. While these detectors can adapt their internal representations, such as cluster structures or density estimations, to reflect changes in data distribution, they do not directly support the adaptation of supervised classification models. Instead, they can trigger mechanisms like model freezing, reinitialization, or selective label querying. As such, they play a crucial role in drift-aware learning pipelines, even when full supervision is not available.

Due to the experimental setup employed, our evaluation does not include deep learning-based concept drift detection methods. Such approaches typically require specialized hardware and tailored software environments, which were beyond the scope of this investigation. As a result, our evaluation is limited to traditional, non-deep learning detectors, potentially overlooking insights that could arise from deep neural architectures. This limitation should be considered when interpreting our results, as it may restrict their generalizability to scenarios where deep learning-based methods are applicable.

Ultimately, our experimental results underscore that concept drift detection in delayed and partially labeled data streams remains a largely underexplored and open research challenge. This highlights the need for novel strategies that effectively integrate semi-supervised learning with mechanisms for assessing the relevance of delayed labels in the presence of evolving data characteristics. Such integration has the potential to open up new and promising directions for future research in this emerging area.

Data availability

Data and code were shared in the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by CNPq (Brazilian National Council for Scientific and Technological Development) grants [404771/2024-6, 406354/2023-5, 312755/2023-6, 313053/2023-5], UFBA/CNPq 68/2022 - MAI/DAI, Maria Emilia Foundation grant to 01/2023, INCITE FAPESB (Bahia Research Foundation) grants to PIE0002/2022 and to PIE0007/2026, CAPES (Coordination for the Improvement of Higher Education Personnel – Brazil), and FAPESB grant [1589/2021].

Appendix A. Partially labeled 99% with gradual concept drift

Table A.1
Metrics: TPD, TDD, and TDDA. Partially labeled (99%) with gradual CD. Data Stream Size 100k.

Drift Type and Metric	Data Stream	DDM $HDDM_A$	EDDM $HDDM_W$	ADWIN $RDDM_{30}$	ECDD RDDM	STEPD Test_Ks	SeqDrift2 Test_Wrs	SEED Test_Tt	Multivariate Studd
	$AGRAW_1$	1.00 (1.00)	4.00 (1.75)	1.00 (1.00)	2.00 (1.00)	1.50 (1.00)	1.00 (0.00)	1.00 (0.00)	0.00 (1.00)
		1.00 (1.00)	2.00 (1.00)	1.00 (1.00)	2.00 (0.00)	2.50 (1.00)	**4.00 (0.00)**	2.00 (0.00)	0.00 (0.00)
	$AGRAW_2$	0.00 (2.00)	3.00 (1.00)	0.00 (0.00)	3.00 (1.00)	2.50 (1.00)	0.00 (1.00)	1.00 (1.00)	0.00 (1.00)
		1.00 (2.00)	2.50 (1.00)	0.00 (1.75)	2.00 (2.50)	2.00 (1.00)	**4.00 (0.00)**	2.00 (1.00)	0.00 (0.00)
	LED	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (1.00)	3.00 (2.00)	2.00 (1.00)	3.00 (1.00)	0.00 (0.00)
		0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	2.00 (1.00)	**4.00 (0.00)**	2.00 (2.00)	0.00 (0.00)
GRADUAL TPD	**MIXED**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	3.00 (1.00)	**4.00 (0.00)**	0.00 (0.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	3.00 (1.75)	**4.00 (0.00)**	4.00 (1.00)	0.00 (0.00)
	RandRBF	**4.00 (0.00)**	**4.00 (0.00)**	1.00 (0.00)	**4.00 (0.00)**	4.00 (1.00)	2.00 (0.00)	2.00 (1.00)	0.00 (0.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	3.00 (1.00)	**4.00 (0.00)**	3.00 (2.00)	0.00 (0.00)
	SINE	3.00 (1.00)	4.00 (1.00)	3.00 (1.00)	4.00 (1.00)	4.00 (1.00)	2.00 (1.00)	3.50 (1.00)	0.00 (0.00)
		4.00 (1.00)	4.00 (1.00)	3.00 (1.00)	**4.00 (0.00)**	3.00 (1.75)	**4.00 (0.00)**	3.00 (0.00)	0.00 (0.00)
	WAVEE	0.00 (0.00)	1.00 (2.00)	0.00 (0.00)	1.00 (1.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
		0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	2.00 (1.00)	**4.00 (0.00)**	2.00 (1.75)	0.00 (0.00)
	AGRAW1	**4.00 (1.00)**	19.00 (17.25)	19.00 (2.75)	130.00 (12.75)	32.50 (5.75)	26.00 (5.75)	63.00 (5.50)	5.00 (5.00)
		5.00 (0.75)	25.00 (6.75)	5.00 (1.00)	6.00 (1.75)	292.00 (30.75)	1553.50 (41.50)	258.50 (23.50)	0.00 (0.00)
	AGRAW2	6.00 (2.00)	16.00 (7.50)	18.00 (3.75)	145.00 (9.75)	20.00 (6.75)	13.00 (2.00)	45.00 (8.25)	5.00 (5.00)
		4.00 (0.00)	9.00 (6.75)	5.00 (0.00)	5.00 (1.00)	145.50 (22.00)	2552.50 (26.50)	137.50 (16.25)	0.00 (0.00)
	LED	2.00 (1.75)	0.00 (0.00)	5.00 (1.00)	69.00 (11.25)	84.50 (53.25)	10053.00 (7513.50)	406.00 (61.25)	0.00 (0.00)
		2.00 (1.00)	20.50 (10.25)	**2.00 (0.00)**	**2.00 (0.00)**	315.00 (45.50)	2087.50 (64.25)	285.00 (29.00)	0.00 (0.00)
GRADUAL TDD	**MIXED**	**4.00 (0.00)**	19.00 (13.00)	22.00 (4.00)	111.50 (17.25)	37.00 (6.75)	12.00 (2.00)	44.50 (7.75)	3.00 (2.00)
		4.00 (0.75)	4.00 (0.75)	6.00 (3.75)	8.50 (2.75)	67.00 (18.50)	2423.00 (45.00)	58.50 (11.75)	0.00 (0.00)
	RandRBF	**4.00 (0.00)**	9.00 (2.00)	17.00 (1.75)	63.00 (4.75)	47.50 (15.50)	10111.00 (59.00)	509.00 (21.50)	0.00 (0.00)
		4.00 (0.00)	**4.00 (0.00)**	4.00 (1.00)	5.00 (1.00)	145.50 (17.00)	2379.50 (45.75)	129.00 (26.25)	2.50 (1.75)
	SINE	5.00 (2.00)	29.00 (16.50)	22.00 (3.00)	146.00 (18.75)	26.50 (8.75)	11.00 (3.00)	64.50 (6.75)	1.00 (0.75)
		4.00 (1.00)	**4.00 (0.00)**	6.00 (1.00)	7.00 (2.75)	84.00 (14.25)	2537.00 (42.00)	81.00 (22.00)	0.00 (0.00)
	WAVEE	0.00 (1.00)	3.00 (8.00)	0.00 (1.50)	155.00 (14.25)	27.00 (12.75)	0.00 (0.00)	9.00 (4.00)	0.00 (0.00)
		0.50 (1.75)	10.00 (3.00)	**4.00 (6.75)**	8.50 (6.75)	230.50 (43.00)	2479.00 (61.00)	275.00 (44.00)	0.00 (0.00)
	AGRAW1	10347.75 (2071.94)	2053.41 (1293.57)	2108.63 (283.42)	26.59 (53.87)	561.86 (313.85)	1172.22 (482.52)	87.94 (86.29)	3023.27 (2900.08)
		7535.60 (1734.66)	1237.42 (562.16)	8288.60 (2097.33)	6832.50 (2086.37)	4.47 (5.50)	**0.11** (0.05)	6.68 (7.79)	0.00 (0.00)
	AGRAW2	6764.09 (2309.07)	2249.06 (1327.53)	2182.22 (457.08)	14.98 (11.67)	1138.95 (998.30)	2671.16 (825.35)	357.19 (154.32)	3155.68 (2984.79)
		9407.00 (462.56)	3425.22 (5333.82)	8132.70 (34.05)	8039.20 (1338.90)	80.58 (16.06)	**0.04 (0.02)**	82.06 (16.60)	0.00 (0.00)
	LED	4286.50 (3000.50)	0.00 (0.00)	1401.60 (348.80)	49.08 (44.30)	129.32 (165.71)	2.29 (184.97)	2.54 (1.47)	0.00 (0.00)
		2401.00 (1632.75)	331.96 (459.03)	4231.25 (509.25)	3769.00 (153.50)	2.61 (3.15)	**0.04 (0.03)**	3.94 (2.37)	0.00 (0.00)
GRADUAL TDDA	**MIXED**	9979.50 (28.12)	2169.87 (2746.19)	1784.73 (306.63)	12.92 (16.12)	146.28 (175.69)	3283.33 (553.85)	312.64 (237.49)	5524.50 (4328.69)
		9184.00 (1861.92)	9909.75 (1516.05)	6655.41 (4492.67)	4675.28 (1268.95)	82.30 (96.31)	**0.04 (0.02)**	105.03 (162.26)	0.00 (0.00)
	RandRBF	10035.88 (17.69)	4707.44 (1100.43)	2328.47 (239.07)	48.48 (25.27)	305.31 (133.32)	2.99 (1.93)	1.42 (0.85)	0.00 (0.00)
		8510.12 (1599.31)	9920.00 (31.19)	10040.00 (2008.26)	7974.60 (1998.16)	27.09 (24.60)	**0.05 (0.05)**	46.45 (45.32)	10655.50 (1856.50)
	SINE	7997.50 (3343.81)	1430.95 (839.44)	1788.37 (238.91)	12.88 (11.02)	932.90 (411.55)	3609.09 (925.39)	124.54 (102.45)	15323.50 (2831.00)
		9422.25 (1856.95)	9943.88 (39.56)	6667.08 (1326.12)	5661.85 (2059.94)	235.95 (87.50)	**0.04 (0.02)**	252.66 (79.44)	0.00 (0.00)
	WAVEE	0.00 (21.00)	0.00 (115.72)	0.00 (0.00)	10.00 (13.39)	266.93 (421.76)	0.00 (0.00)	1793.29 (1533.13)	0.00 (0.00)
		0.00 (1890.75)	1634.75 (2519.58)	2030.04 (2713.02)	2518.82 (2456.07)	5.66 (8.69)	**0.03 (0.03)**	3.96 (7.54)	0.00 (0.00)

Appendix B. Delay Labels 10.000 with gradual concept drift

Table B.1 shows the results for the experiment with gradual concept drift. Similar to the previous experiment, most detectors achieved high TPD values, although the Multivariate and Studd detectors had zero TPD values in some data streams. For TDD, Test_Wrs again presented the highest values. Consequently, as with the abrupt drift experiment, Test_Wrs showed the lowest TDDA values, consistent with the earlier analysis.

Table B.1
Metrics: TPD, TDD, and TDDA. Maximum Delay Labels 10.000 with gradual CD. Data Stream Size 100k.

Drift Type and Metric	Data Stream	DDM $HDDM_A$	EDDM $HDDM_W$	ADWIN $RDDM_{30}$	ECDD RDDM	STEPD Test_Ks	SeqDrift2 Test_Wrs	SEED Test_Tt	Multivariate Studd
	AGRAW$_1$	**4.00 (0.00)**	3.00 (2.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	2.00 (2.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
	AGRAW$_1$	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	2.00 (2.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
	LED	2.00 (1.00)	0.00 (0.00)	2.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	2.00 (0.75)	**4.00 (0.00)**	0.00 (0.00)
		2.00 (0.00)	**4.00 (0.00)**	2.00 (0.00)	2.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
GRADUAL TPD	**MIXED**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	2.00 (0.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
	RandRBF	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	2.00 (1.75)
	SINE	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	1.00 (0.00)
		4.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
	WAVEE	0.00 (0.75)	0.00 (1.00)	0.00 (0.00)	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)	3.00 (1.00)	0.00 (0.00)
		0.00 (1.00)	**4.00 (0.00)**	2.00 (2.75)	4.00 (1.00)	**4.00 (0.00)**	**4.00 (0.00)**	**4.00 (0.00)**	0.00 (0.00)
	AGRAW1	**4.40 (+0.71)**	20.33 (+12.29)	19.30 (+2.10)	130.30 (+11.15)	32.37 (+5.27)	358.13 (+1794.73)	63.33 (+4.32)	6.07 (+4.75)
		5.00 (+0.86)	23.80 (+4.64)	4.67 (+1.25)	5.83 (+1.37)	293.10 (+20.86)	1888.70 (+1794.21)	256.07 (+18.33)	0.00 (+0.00)
	AGRAW2	6.10 (+1.45)	17.43 (+7.50)	18.13 (+2.46)	145.57 (+10.24)	20.63 (+7.12)	13.27 (+2.21)	45.13 (+6.18)	6.03 (+4.75)
		4.33 (+0.79)	8.53 (+3.43)	4.93 (+0.73)	5.47 (+0.88)	148.33 (+13.63)	2884.27 (+1796.83)	137.23 (+13.18)	0.00 (+0.00)
	LED	1.90 (+1.01)	0.00 (+0.00)	**4.73 (+0.96)**	69.27 (+8.37)	94.60 (+44.57)	7386.67 (+4430.73)	372.70 (+71.74)	0.00 (+0.00)
GRADUAL TDD		2.47 (+0.81)	23.13 (+8.11)	2.33 (+0.75)	2.50 (+1.41)	309.97 (+37.60)	2083.27 (+63.02)	286.70 (+28.52)	0.13 (+0.34)
	MIXED	4.50 (+1.12)	18.63 (+9.30)	23.03 (+2.55)	110.67 (+12.84)	37.53 (+6.92)	12.10 (+2.33)	44.23 (+6.25)	3.50 (+1.63)
		4.40 (+0.80)	4.53 (+0.99)	6.10 (+1.70)	9.90 (+3.51)	70.20 (+12.57)	2419.40 (+38.99)	59.57 (+13.95)	0.00 (+0.00)
	RandRBF	4.17 (+0.45)	9.03 (+1.66)	16.77 (+1.12)	62.87 (+3.38)	48.67 (+10.39)	8128.17 (+3996.47)	503.97 (+23.95)	0.00 (+0.00)
		4.20 (+0.40)	**4.10 (+0.40)**	4.43 (+0.62)	4.77 (+0.96)	145.30 (+13.13)	2376.90 (+32.90)	126.43 (+16.89)	2.33 (+0.98)
	SINE	5.20 (+1.42)	30.70 (+11.88)	22.33 (+2.24)	146.77 (+12.47)	28.17 (+7.24)	344.50 (+1794.84)	65.50 (+5.36)	1.27 (+0.44)
		4.33 (+0.54)	4.13 (+0.34)	5.90 (+0.70)	8.03 (+2.55)	417.80 (+1792.41)	2543.10 (+29.27)	85.10 (+16.33)	0.00 (+0.00)
	WAVEE	0.50 (+0.96)	6.47 (+8.52)	0.97 (+1.80)	153.60 (+16.70)	28.67 (+8.58)	0.47 (+0.88)	9.07 (+3.96)	0.00 (+0.00)
		0.93 (+1.15)	10.37 (+3.27)	**5.03 (+3.97)**	8.90 (+4.96)	229.80 (+30.15)	2799.90 (+1808.59)	264.30 (+29.24)	0.00 (+0.00)
	AGRAW1	9627.03 (+1231.40)	2232.04 (+899.57)	2096.93 (+220.70)	41.67 (+34.21)	584.95 (+229.79)	1195.05 (+438.95)	101.54 (+60.91)	3912.26 (+3201.21)
		7402.51 (+1477.85)	1298.49 (+446.81)	8678.26 (+1617.45)	7350.70 (+1575.55)	7.41 (+8.45)	**0.11 (+0.05)**	9.73 (+8.97)	0.00 (+0.00)
	AGRAW2	7049.59 (+1723.04)	2571.90 (+1280.94)	2232.22 (+326.78)	16.29 (+7.28)	1370.67 (+916.81)	2717.91 (+604.71)	359.38 (+121.86)	3938.87 (+3206.63)
		8789.42 (+1442.50)	5253.12 (+2996.14)	8401.85 (+1141.73)	7441.85 (+1097.02)	81.53 (+20.81)	**0.04 (+0.02)**	89.05 (+17.40)	0.00 (+0.00)
	LED	4382.53 (+1791.10)	0.00 (+0.00)	1546.34 (+339.99)	52.84 (+39.93)	178.98 (+183.87)	133.61 (+226.92)	8.65 (+15.43)	0.00 (+0.00)
GRADUAL TDDA		3028.68 (+3519.45)	654.17 (+801.51)	4162.30 (+895.02)	3920.83 (+1333.79)	3.09 (+2.57)	**0.05 (+0.03)**	4.01 (+2.28)	380.17 (+1345.48)
	MIXED	9233.53 (+1488.37)	3060.63 (+2444.36)	1724.44 (+183.00)	16.60 (+12.43)	227.46 (+299.06)	3388.55 (+658.04)	363.27 (+188.53)	5828.07 (+3800.27)
		8549.19 (+1208.21)	9069.18 (+1482.77)	7098.67 (+2027.73)	4214.79 (+1379.37)	94.76 (+75.13)	**0.04 (+0.02)**	158.97 (+142.04)	0.00 (+0.00)
	RandRBF	9699.74 (+926.90)	4677.45 (+833.54)	2367.15 (+156.87)	52.87 (+22.67)	354.32 (+192.15)	47.07 (+121.08)	2.97 (+5.81)	0.00 (+0.00)
		8283.71 (+960.92)	9732.76 (+736.12)	9217.98 (+1131.10)	8578.42 (+1546.72)	31.74 (+18.97)	**0.05 (+0.03)**	55.80 (+33.98)	10524.06 (+2764.75)
	SINE	8177.24 (+1838.21)	1590.54 (+786.74)	1776.26 (+173.44)	14.54 (+9.56)	910.02 (+381.21)	3508.99 (+892.88)	133.39 (+71.03)	13748.72 (+3025.13)
		8850.90 (+946.62)	9675.04 (+681.27)	6875.37 (+817.99)	5352.87 (+1351.64)	233.27 (+71.43)	**0.04 (+0.01)**	249.17 (+62.84)	0.00 (+0.00)
	WAVEE	819.17 (+1713.06)	802.23 (+3379.66)	172.93 (+684.04)	13.58 (+12.46)	413.69 (+380.92)	502.22 (+1741.75)	2236.01 (+1503.19)	0.00 (+0.00)
		1069.97 (+2350.06)	2274.46 (+2456.87)	2285.61 (+1822.89)	2586.02 (+1622.56)	7.85 (+6.63)	**0.04 (+0.05)**	6.49 (+6.54)	0.00 (+0.00)

Appendix C. Experiment results complementary to those presented in the article

The experiments were conducted using an experimental setup with partially labeled datasets at 95%, 96%, and 97%. Each percentage exhibited both abrupt and gradual concept drift. The evaluated metrics included TPD, TDD, and TDDA. The results were obtained using the following data generators: Agrawal, LED, Mixed, Random RBF, Sine, and Waveform.

The results related to the delay labeled were omitted because they presented similar conclusions to the partially labeled data stream. The complete results of the entire experimental set can be found in the following repository: https://github.com/BrennoMello/comparation-partially-delay-drift.

References

[1] J. Gama, Ž. Indrė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (4) (2014) 44.

[2] H.M. Gomes, M. Grzenda, R. Mello, J. Read, M.H. Le Nguyen, A. Bifet, A survey on semi-supervised learning for delayed partially labelled data streams, ACM Comput. Surv. 55 (4) (2022) 1–42.

[3] C. Fahy, S. Yang, M. Gongora, Scarcity of labels in non-stationary data streams: a survey, ACM Comput. Surv. 55 (2) (2022) 1–39.

[4] N. Gunasekara, B. Pfahringer, H.M. Gomes, A. Bifet, Survey on online streaming continual learning, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI '23, 2023. https://doi.org/10.24963/ijcai.2023/743

[5] N. Gunasekara, B. Pfahringer, H. Murilo Gomes, A. Bifet, Y.S. Koh, Recurrent concept drifts on data streams, in: K. Larson (Ed.), Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24, International Joint Conferences on Artificial Intelligence Organization, 2024, pp. 8029–8037. Survey Track. https://doi.org/10.24963/ijcai.2024/888

[6] I. Žliobaite, Change with delayed labeling: when is it detectable?, in: 2010 IEEE International Conference on Data Mining Workshops, IEEE, 2010, pp. 843–850.

[7] E.S. Page, Continuous inspection schemes, Biometrika 41 (1/2) (1954) 100–115.

[8] 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.

[9] A. Bifet, R. Gavalda, Learning from time-changing data with adaptive windowing, in: Proceedings of the 2007 SIAM International Conference on Data Mining, SIAM, 2007, pp. 443–448.

[10] J. Gama, P. Medas, G. Castillo, P. Rodrigues, Learning with drift detection, in: Advances in Artificial Intelligence–SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings 17, Springer, 2004, pp. 286–295.

[11] M. Baena-Garcıa, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, R. Morales-Bueno, Early drift detection method, in: Fourth International Workshop on Knowledge Discovery from Data Streams, vol. 6, Citeseer, 2006, pp. 77–86.

[12] I. Frias-Blanco, J. del Campo-Ávila, 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) (2014) 810–823.

[13] K. Nishida, K. Yamauchi, Detecting concept drift using statistical testing, in: International Conference on Discovery Science, Springer, 2007, pp. 264–269.

[14] V.M.A. Souza, F.A. Chowdhury, A. Mueen, Unsupervised drift detection on high-speed data streams, in: 2020 IEEE International Conference on Big Data (Big Data), IEEE, 2020, pp. 102–111.

[15] J. Klikowski, Concept drift detector based on centroid distance analysis, in: 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, 2022, pp. 1–8.

[16] W.J. Faithfull, J.J. Rodríguez, L.I. Kuncheva, Combining univariate approaches for ensemble change detection in multivariate data, Inf. Fusion 45 (2019) 202–214.

[17] Z. Wang, W. Wang, Concept drift detection based on Kolmogorov–Smirnov test, in: Artificial Intelligence in China: Proceedings of the International Conference on Artificial Intelligence in China, Springer, 2020, pp. 273–280.

[18] K.T. Jafseer, S. Shailesh, A. Sreekumar, Modeling concept drift detection as machine learning model using overlapping window and Kolmogorov–Smirnov test, in: Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND 2021, Springer, 2023, pp. 113–129.

[19] P. Porwik, B.M. Dadzie, Detection of data drift in a two-dimensional stream using the Kolmogorov-Smirnov test, Procedia Comput. Sci. 207 (2022) 168–175.

[20] D.M. Dos Reis, P. Flach, S. Matwin, G. Batista, Fast unsupervised online drift detection using incremental kolmogorov-smirnov test, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1545–1554.

[21] Ö. Gözüaçık, A. Büyükçakır, H. Bonab, F. Can, Unsupervised concept drift detection with a discriminative classifier, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2365–2368.

[22] J. Xuan, J. Lu, G. Zhang, Bayesian nonparametric unsupervised concept drift detection for data stream mining, ACM Trans. Intell. Syst. Technol. 12 (1) (2020) 1–22.

[23] H. Kingetsu, K. Kobayashi, Born-again decision boundary: unsupervised concept drift detection by inspector neural network, in: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8.

[24] M. Fuccellaro, L. Simon, A. Zemmari, Partially supervised classification for early concept drift detection, in: 2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2022, pp. 190–196.

[25] Ö. Gözüaçık, F. Can, Concept learning using one-class classifiers for implicit drift detection in evolving data streams, Artif. Intell. Rev. 54 (5) (2021) 3725–3747.

[26] S. Zheng, S.B. van der Zon, M. Pechenizkiy, C.P. de Campos, W. van Ipenburg, H. de Harder, R. Nederland, Labelless concept drift detection and explanation, in: NeurIPS 2019 Workshop on Robust AI in Financial Services: Data, Fairness, Explainability, Trustworthiness, and Privacy, 2019.

[27] Y. Xu, D. Klabjan, Concept drift and covariate shift detection ensemble with lagged labels, in: 2021 IEEE International Conference on Big Data (Big Data), IEEE, 2021, pp. 1504–1513.

[28] F.A. Melo, A.C. de Carvalho, A.C. Lorena, L.P.F. Garcia, Model performance prediction: a meta-Learning approach for concept drift detection, in: International Conference on Hybrid Artificial Intelligence Systems, Springer, 2023, pp. 51–62.

[29] P. Lindstrom, B. Mac Namee, S.J. Delany, Drift detection using uncertainty distribution divergence, Evol. Syst. 4 (2013) 13–25.

[30] Y.I. Kim, C.H. Park, Concept drift detection on streaming data under limited labeling, in: 2016 IEEE International Conference on Computer and Information Technology (CIT), IEEE, 2016, pp. 273–280.

[31] T.S. Sethi, M. Kantardzic, On the reliable detection of concept drift from streaming unlabeled data, Expert Syst. Appl. 82 (2017) 77–99.

[32] M. Heyden, E. Fouché, V. Arzamasov, T. Fenn, F. Kalinke, K. Böhm, Adaptive Bernstein change detector for high-dimensional data streams, Data Min. Knowl. Discov. 38 (3) (2024) 1334–1363.

[33] H. Yu, Q. Zhang, T. Liu, J. Lu, Y. Wen, G. Zhang, Meta-ADD: a meta-learning based pre-trained model for concept drift active detection, Inf. Sci. 608 (2022) 996–1009. https://doi.org/https://doi.org/10.1016/j.ins.2022.07.022

[34] P. Wang, H. Yu, N. Jin, D. Davies, W.L. Woo, QuadCDD: a quadruple-based approach for understanding concept drift in data streams, Expert Syst. Appl. 238 (2024) 122114. https://doi.org/https://doi.org/10.1016/j.eswa.2023.122114

[35] V. Cerqueira, H.M. Gomes, A. Bifet, Unsupervised concept drift detection using a student–teacher approach, in: Discovery Science: 23rd International Conference, DS 2020, Thessaloniki, Greece, October 19–21, 2020, Proceedings 23, Springer, 2020, pp. 190–204.

[36] V. Cerqueira, H.M. Gomes, A. Bifet, L. Torgo, STUDD: A student–teacher method for unsupervised concept drift detection, Mach. Learn. 112 (11) (2023) 4351–4378.

[37] E.B. Gulcan, F. Can, Unsupervised concept drift detection for multi-label data streams, Artif. Intell. Rev. 56 (3) (2023) 2401–2434.

[38] J. Gama, R. Sebastiao, P.P. Rodrigues, Issues in evaluation of stream learning algorithms, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 329–338.

[39] M. Grzenda, H.M. Gomes, A. Bifet, Delayed labelling evaluation for data streams, Data Min. Knowl. Discov. 34 (5) (2020) 1237–1266.

[40] M. Grzenda, H.M. Gomes, A. Bifet, Performance measures for evolving predictions under delayed labelling classification, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8.

[41] A. Bifet, G. de Francisci Morales, J. Read, G. Holmes, B. Pfahringer, Efficient online evaluation of big data stream classifiers, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 59–68.

[42] A. Bifet, Classifier concept drift detection and the illusion of progress, in: Artificial Intelligence and Soft Computing: 16th International Conference, ICAISC 2017, Zakopane, Poland, June 11–15, 2017, Proceedings, Part II 16, Springer, 2017, pp. 715–725.

[43] G.I. Webb, R. Hyde, H. Cao, H.L. Nguyen, F. Petitjean, Characterizing concept drift, Data Min. Knowl. Discov. 30 (4) (2016) 964–994.

[44] P. Domingos, G. Hulten, Mining high-speed data streams, in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, pp. 71–80.