Challenges in benchmarking stream learning algorithms with real-world data

Vinicius M. A. Souza[1,2] · Denis M. dos Reis[1] · André G. Maletzke[1] · Gustavo E. A. P. A. Batista[1,3]

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2020

Abstract

Streaming data are increasingly present in real-world applications such as sensor measurements, satellite data feed, stock market, and financial data. The main characteristics of these applications are the online arrival of data observations at high speed and the susceptibility to changes in the data distributions due to the dynamic nature of real environments. The data stream mining community still faces some primary challenges and difficulties related to the comparison and evaluation of new proposals, mainly due to the lack of publicly available high quality non-stationary real-world datasets. The comparison of stream algorithms proposed in the literature is not an easy task, as authors do not always follow the same recommendations, experimental evaluation procedures, datasets, and assumptions. In this paper, we mitigate problems related to the choice of datasets in the experimental evaluation of stream classifiers and drift detectors. To that end, we propose a new public data repository for benchmarking stream algorithms with real-world data. This repository contains the most popular datasets from literature and new datasets related to a highly relevant public health problem that involves the recognition of disease vector insects using optical sensors. The main advantage of these new datasets is the prior knowledge of their characteristics and patterns of changes to adequately evaluate new adaptive algorithms. We also present an in-depth discussion about the characteristics, reasons, and issues that lead to different types of changes in data distribution, as well as a critical review of common problems concerning the current benchmark datasets available in the literature.

Keywords Data stream · Concept drift · Classification · Drift detection · Benchmark data

---

✉ Vinicius M. A. Souza
vinicius@unm.edu

Extended author information available on the last page of the article

1 Introduction

In the last 20 years, we have witnessed the emergence and notable increase in the interest of algorithms that learn from streaming data. This new generation of machine learning methods is designed to deal with continuous flows of data. Frequently, such streams comprise changes in the distribution of data, which are governed by the dynamics of real-world problems and application domains that evolve. In the context of machine learning, these changes in data distribution are named *concept drifts* (Widmer and Kubat 1996) and typically occur in data that are observed continuously at a fast rate, which in turn impose time and memory constraints on the algorithms that process them.

Batch learning is a standard approach for machine learning, where a static model is built offline on the whole dataset at once, and never updated in the future. Batch learning is a mature field with clear procedures to evaluate and compare different methods using a vastitude of data shared by researchers for benchmarking. However, in the online scenario of data stream mining, we still face some primary challenges and difficulties related to the comparison and evaluation of new proposals, mainly due to the lack of publicly available high quality non-stationary real-world datasets. For example, we found more than 300 stationary datasets for classification problems from different domains in the UCI Machine Learning Repository (Dua and Graff 2017). In particular, for time-series classification, the UEA & UCR Time Series Classification Repository (Bagnall et al. 2019) stores more than 100 datasets and 20 algorithms from the literature. For data stream mining, although there is the popular open-source framework MOA (Bifet et al. 2010a) with a collection of algorithms and synthetic data generators, we do not have any public repository with a reasonably-sized collection of real-world stream datasets accompanied of their detailed description. More alarming, different data stream algorithms often run on specific assumptions about the data (for instance, methods may assume changes to be either incremental or abrupt). However, frequently it is not clear whether employed datasets fulfill such assumptions or not.

As recently noted by Krawczyk et al. (2017), the comparison of stream learning algorithms is not an easy task, as authors do not always follow the same recommendations, experimental evaluation procedures, datasets, and assumptions. In this paper, we want to mitigate the problems related to the choice of datasets in the experimental evaluation of stream classifiers and drift detectors.

In summary, the main contributions of this article are the following:

– Presentation of basic concepts of data stream mining accompanied by an in-depth discussion about the characteristics, reasons, and issues, which lead to different types of changes in data distribution;
– A review of the main real-world datasets adopted in the evaluation of stream learning methods accompanied by a critical discussion of issues concerning the data and the challenges imposed due to the lack of a benchmark standard;
– Presentation of a relevant public health problem that involves the recognition of disease vector insects by an optical sensor, which is responsible for generating evolving data over time;

– Building (data collection, preprocessing, and features extraction) of 11 new real-world datasets with controlled concept changes where it is possible to identify the types/patterns of changes and when they occur for each dataset. Such data are accompanied by an experimental evaluation that includes state-of-the-art classifiers and drift detectors;
– Development and availability of a repository[1] with 27 real-world datasets for benchmarking the evaluation of stream classifiers and change detectors.

This paper is organized as follows. In Sect. 2, we provide a background on data stream and concept drift, as well as a discussion about characteristics, reasons, and issues that lead to changes in data distribution. In Sect. 3, we present an overview of the most common datasets used in the evaluation of stream mining approaches. In Sect. 4, we discuss the challenges faced by the stream mining community when the currently most popular real-world datasets are evaluated. In Sect. 5, we introduce a benchmark dataset for stream learning with controlled concept drifts, which are generated by an optical sensor that measures characteristics of insect flights. In Sect. 6, we present to the stream learning community a new public repository with an initial amount of 27 datasets. In Sect. 7, we perform an evaluation followed by a discussion concerning the 11 datasets introduced in this paper. Finally, we conclude our work in Sect. 8.

2 Background

In this section, we present the main concepts and definitions regarding data streams, concept drift and the tasks of classification and drift detection under non-stationary environments.

2.1 Data stream

A data stream is an ordered sequence of instances continuously observed over time. Streaming data are increasingly prevalent in real-world applications. Representative examples include network traffic, database transactions, sensor measurements, satellite data feed, stock market and financial data, georeferenced data from mobile devices, among others.

Formally, a data stream is a sequence of instances $\mathcal{DS} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_t, \ldots\}$, where $\mathbf{X}_t \in \mathcal{X}$ is a d-dimensional vector in the feature space that was observed at time t. In practice, $\mathbf{X}_t$ is an ordered list of descriptive attributes that represent the observation being made. The attributes qualitative (nominal, ordinal, or binary) or quantitative (discrete or continuous).

Among possible tasks such as clustering, regression, graph-mining, outlier detection, and recommender systems, classification is probably the most prominent task on data streams and the focus of this work. In classification problems, each instance $\mathbf{X}_t \in \mathcal{X}$ is associated to a class label $y_t \in \mathcal{Y}$, where $\mathcal{Y}$ contains l possible labels, $\mathcal{Y} = \{C_1, C_2, \ldots, C_l\}$. Therefore, the data stream of classification problems can be seen as a set of pairs $\mathcal{DS} = \{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \ldots, (\mathbf{X}_t, y_t), \ldots\}$.

---

[1] USP Data Stream Repository—Available online at https://sites.google.com/view/uspdsrepository.

Due to a data stream's potentially infinite length, traditional batch methods are often not applicable (Gama and Gaber 2007). These methods typically fail to comply with at least one of the three most prominent restrictions of the stream setting (Bifet 2009):

1. In general, it is impractical to store all events of a stream due to its potentially infinite length. Only a small portion can be retained in memory;
2. Fast-paced streams require each event to be processed in real-time and, afterward, discarded; and
3. The underlying distribution of the data may change over time. Hence, old data can become irrelevant or even detrimental to model the current concept. We note that, contrary to batch learning, in data stream we expect the characteristics of the newly observed data to change when compared to past data.

The first constraint limits the amount of memory the algorithms can use, and the second constraint limits the time that is available for processing each event. Therefore, the first two restrictions lead to the development of techniques that reduce the information in a stream of data, such as sampling (Chaudhuri et al. 1999), *sketching* (Alon et al. 1999), histograms (Gilbert et al. 2002), *wavelets* (Matias et al. 2000), and sliding windows (Datar et al. 2002). The third constraint guides the development of algorithms that are capable of detecting changes in the data and reacting by updating the existing models.

The non-stationarity of many real environments may lead to changes in the underlying distribution of the observed data, a phenomenon that goes by many names in literature, where the most common is *concept drift* (Widmer and Kubat 1996). According to the same terminology, the data distribution in a given moment is called *concept*. Additionally, a change in its parameters is a *drift*.

Drifts constitute a central issue since they can decrease the performance of machine learning models induced with historical data (Pan and Yang 2009; Quionero-Candela et al. 2009; Saenko et al. 2010; Ben-David et al. 2007). A closely related problem in batch learning is *concept shift*, which occurs when a model is trained with data from one distribution and is later applied on data that follow a different distribution (Moreno-Torres et al. 2012). Similarly, *transfer learning* aims to extract the knowledge from a source domain where abundant labeled data are available and applies this knowledge to a related target domain in which insufficient labeled data are available (Pan and Yang 2009).

2.2 Concept drift

Concept drifts may manifest with different velocity, severity, and patterns. To illustrate different patterns, consider a concept represented by the color and shape of a geometric figure. Figure 1 illustrates three types of drifts discussed in this work: abrupt, gradual, and incremental.

An abrupt drift occurs when the underlying distribution of the data suddenly changes into a different distribution. In other words, after an abrupt transition between two observations, all new data points belong to a concept different from the previous one. In the incremental change, there are several intermediary concepts between one initial concept and a final concept. Consecutive concepts within this transition period may be

Fig. 1 Representation of three types of concept drift over time

Abrupt

Gradual

Incremental

Time
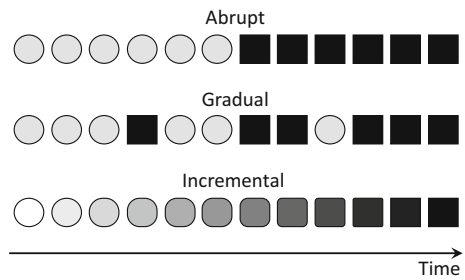
indistinguishable. In the case of gradual concept drift, the transition between two concepts occurs smoothly. However, differently from incremental drift, in a gradual drift, the probability of observing instances that belong to the previous concept decreases over time while, simultaneously, the probability of observing instances that belong to the new concept increases, even though both concepts are remarkably distinct and stationary during the transition period. Concepts that were seen in the past and are later observed again are called recurring concepts. We note that one-off random deviations in the data, such as outliers or noise, are not considered to be concept drifts.

A practical way to identify such patterns of change is to analyze the data distribution over sliding windows in the stream. A window represents a sample of examples that are observed in sequence within a period. When we move the boundaries that define the first and last data points of this window over the stream to comprise different intervals, we have a sliding window. We note that the use of windows imposes the choice of essential parameters. The most common is the number of data points that will be comprised by the window, and how much is the overlap between consecutive windows.

We can only indirectly observe the underlying concept of the data by analyzing samples of instances in the sliding window. Therefore, the observation of concept drift is also indirect. As the number of instances is finite, there is a discrete and finite number of observable distributions that can be analyzed. Note that which examples are included in the window change the perception of the distribution: a bigger window can hide inner distributions that would be perceived as distinct with smaller windows. On the other hand, it may be infeasible to recognize certain concepts if we can only observe too few data points for each occurring concept.

2.3 Independent distribution

In batch learning, a common assumption is that examples are independent and identically distributed (i.i.d.). In data stream applications, such an assumption usually does not conform to reality. Identically distributed means that the joint distribution of an example and its class label is the same at any time, that is, $P(\mathbf{X}_t, y_t) = P(\mathbf{X}_{t'}, y_{t'})$, when $t \neq t'$. Meanwhile, independently distributed means that the probability of the current label does not depend on what was observed before, that is, $P(y_t) = P(y_t | y_{t-k})$.

Since data streams are generated in dynamic environments, examples are not identically distributed due to the occurrence of concept drifts. Additionally, while most of the literature assumes independence between examples, some recent studies have found a significant *temporal dependence* in many real-world data (Bifet et al. 2013; Zliobaite et al. 2015). This dependence on historical class labels has a direct impact on the design and evaluation of stream approaches.

The existence of temporal dependence reinforces that the line separating data stream from time-series is blurred. One possible view for streams is that the instances are independent of each other in the sense that the occurrence or absence of one *particular* instance implies neither the presence or absence of other *particular* instances (Reis et al. 2018a). However, the observation of any instance is under the influence of a shared background concept, which is not directly observable. Even though the instances are not particularly dependent on each other, the occurrence of one instance *may* be indicative of the likelihood of observing instances of a particular class or in a specific region of the feature space, due to this common background concept. One example is motion recognition. In this problem, sensors attached to a particular participant may indicate that this person is performing the *eating* activity. Since this assessment can be an indication that it is lunchtime, the likelihood of observing other people performing the same activity may increase, albeit all the analyzed people being unrelated.

A different view, closer to time-series, is that one attribute value is the result of an auto-regressive transformation applied to previous instances. Examples are the variation in the price of commodities such as electricity (Zliobaite 2013) and the evolution of weather (Ditzler and Polikar 2013).

Datasets can mix the two views mentioned above by combining features from different sources. In this case, it is imperative to make the distinction between both views, since change detection in time-series and drift detection in data streams with independent examples are distinct research topics that require remarkably different approaches. To elucidate, consider a problem where one of the descriptive features is a time-series defined by a strictly crescent monotonic function. Any two non-overlapping sliding windows over this series have statistically different probability distributions. Although this difference in the distribution exists for the time-series, it may not be indicative of a change in other aspects of data. One example is the analysis of the behavior of fish species. Any consistent change in water temperature is statistically detected. However, the magnitude of this change may not be enough to affect fish behavior.

2.4 Types of concept drift

In the classification task, a predictive model learns a function f that maps the input variables ($\mathcal{X}$) representing the feature space to discrete output variables ($\mathcal{Y}$) of class labels. Fawcett and Flach (2005) state that there are two types of problems based on the causal direction of such a relationship between the feature space and the class labels. Additionally, only some types of drifts can occur for each type of problem. The types of problems are:

$\mathcal{X} \to \mathcal{Y}$: the class label is derived from the behavior of the instance. One example would be recognizing specific body movements of a person with sensors. The joint distribution is often written as $P(X, Y) = P(Y|X)P(X)$;

$\mathcal{Y} \to \mathcal{X}$: the class label determines the values of the features. One example would be a disease diagnosis in which the disease would cause symptoms. The joint distribution is often written as $P(X, Y) = P(X|Y)P(Y)$.

Furthermore, drift incurs a difference between two concepts. To simplify our discussion, we call P_A the probability before the drift and P_B the probability after the drift has occurred so that we can compare both distributions.

Based on the types of problems listed before, Moreno-Torres et al. (2012) review and compile the nomenclatures and definitions from literature into a single reference list of types of drifts. Although general changes are commonly referred to as *concept drift* in the data stream literature, Moreno-Torres et al. (2012) provide a normalization in which all types of change go by *dataset shift*. Additionally, a change belongs to one among three more specific types: *covariate shift*, *prior probability shift*, and *concept shift*.

Covariate shift refers to changes in the feature space alone and, according to Moreno-Torres et al. (2012), only happens in $\mathcal{X} \to \mathcal{Y}$ problems. It is defined as follows:

Definition 1 Covariate shift is the case where $P_A(Y|X) = P_B(Y|X)$ and $P_A(X) \neq P_B(X)$.

Prior probability shift refers to changes in the class proportions alone. It is the main subject of study in a new subfield of Machine Learning called class prior estimation or quantification (González et al. 2017). According to Moreno-Torres et al. (2012), this type of change only happens in $\mathcal{Y} \to \mathcal{X}$ problems, and is defined as follows:

Definition 2 Prior probability shift is the case where $P_A(X|Y) = P_B(X|Y)$ and $P_A(Y) \neq P_B(Y)$.

We note that, in prior probability shift, although $P_A(X|Y) = P_B(X|Y)$, it is not necessarily true that $P_A(Y|X) = P_B(Y|X)$. This can be easily observed by changing $P(Y)$ in datasets with classes that highly overlap. To illustrate, recall that classifiers typically learn to classify instances in a region of the feature space as the most common class in said region. However, which is the most common class is subject to change according to alterations in $P(Y)$. Yet, the behavior of each class, individually, remains the same.

Concept shift is a change in the relationship between the feature space and the class labels, and is, according to Moreno-Torres et al. (2012), the hardest type of shift. It is defined as follows:

Definition 3 Concept shift is the case where one of the following happens:

1. $P_A(Y|X) \neq P_B(Y|X)$ and $P_A(X) = P_B(X)$ in $\mathcal{X} \to \mathcal{Y}$ problems;
2. $P_A(X|Y) \neq P_B(X|Y)$ and $P_A(Y) = P_B(Y)$ in $\mathcal{Y} \to \mathcal{X}$ problems;
3. $P_A(Y|X) \neq P_B(Y|X)$ and $P_A(X) \neq P_B(X)$ in $\mathcal{X} \to \mathcal{Y}$ problems;
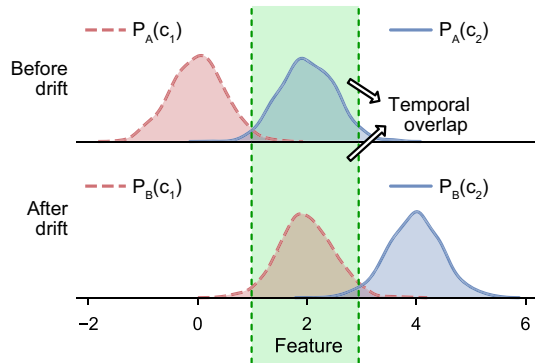4. $P_A(X|Y) \neq P_B(X|Y)$ and $P_A(Y) \neq P_B(Y)$ in $\mathcal{Y} \to \mathcal{X}$ problems.

Fig. 2 Illustration of a case where $P_A(Y|X) \neq P_B(Y|X)$ and $P_A(X) \neq P_B(X)$. There are two classes (c_1 and c_2). Their distributions are shown before and after the drift happened. The drift is a global linear transformation that moved the average feature value two units up. The green shade illustrates a temporal overlap: instances that belong to class c_1 would seem to belong to class c_2 according to the outdated distributions. In this particular example, since there is no change in the proportion of classes, it is also true that $P_A(X|Y) \neq P_B(X|Y)$ and $P_A(Y) = P_B(Y)$. Therefore, this figure represents concepts shifts 2 and 3 (Color figure online)

Condition 1 states that the proportions of the classes, given the characteristics of the data points, change, while the distribution of these characteristics remains the same. In practical terms, we have two effects. First, if we ignore the label information and compare the data before and after the drift, they have the same probability distribution. Second, the proportion of classes in regions of the feature space changes. The only difference between conditions 1 and 3 is that the latter is free of the restriction of preserving $P(X)$.

Condition 2 states that the characteristics that define each class change. However, $P(Y)$ must be preserved, while $P(X)$ can change. $P(Y)$, on its own, dictates the proportion of the classes considering all data points. The only difference between conditions 3 and 4 is that the latter is free of the restriction of preserving $P(Y)$. For that reason, condition 4 is similar to the prior probability shift with the aggravating factor that the characteristics of each class have changed.

Moreno-Torres et al. (2012) state that the concept shifts 3 and 4 are rarer and possibly impossible to tackle. On the other hand, the two first shifts are easier to deal with. However, we find no reason to believe that concept shifts 3 and 4 are rare. We emphasize that such conditions are not mutually exclusive except for the pairs (2, 4) and (1, 3).

A simple global linear transformation that moves all instances towards some direction can cause concept shift 3. In this situation, if the proportion of classes is kept the same after the drift, we would also simultaneously fulfill condition 2. Otherwise, we would simultaneously fulfill condition 4. A linear transformation without changes in the proportion of classes is illustrated in Fig. 2. We analyze a case of temporal overlap on real-world data in Sect. 5.6.

Figure 2 has a *temporal overlap*. The temporal overlap is a superposition of instances that belong to different classes in the feature space, and that only occurs if we ignore
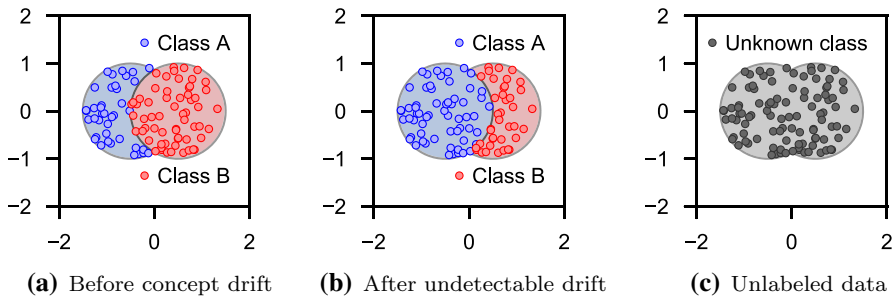
(a) Before concept drift **(b)** After undetectable drift **(c)** Unlabeled data

Fig. 3 Illustration of a concept drift that is undetectable without true labels in a two-dimensional feature space with two classes. Red dots represent events belonging to class A, that are generated with the red-shaded area. Blue dots represent events belonging to class B, that are generated withing the blue-shaded area. In general, undetectable changes are those in which $P_A(X) = P_B(X)$ while $P_A(Y|X) \neq P_B(Y|X)$. Another example is the change of a hyperplane that separates the two classes in the feature space (Reis et al. 2018a) (Color figure online)

the temporal aspect of the data. For example, if we we process the whole dataset at once. If we fail to temporally split the data so that we can separately analyze the concepts before and after the drift, we identify a greater class overlap than the one that exists in each concept individually. In this particular case, our view of the data would suggest that around 50% of c_1 overlaps with around 50% of c_2 in the feature space, which is a strikingly harder classification scenario than the one found in each isolated concept. The existence of temporal overlap reinforces the importance of adequately choosing the parameters of observation windows.

While concept shifts 3 and 4 may be hard to tackle in typical batch learning problems, some assumptions can make them identifiable in streams. For instance, changes in a stream can be incremental and, therefore, traceable over time (Dyer et al. 2014; Souza et al. 2015a, b), or can always lead to a previously seen distribution of the data (Reis et al. 2018b).

Finally, we contest the easiness of concept shift 1: in fact, this type of drift is impossible to detect in unsupervised settings, since we can only observe $P(X)$ and it does not change (Zliobaite 2010). This fact is visually illustrated in Fig. 3. Since the proportion of the classes is changed non-uniformly to preserve $P(X)$, this figure also illustrates a concept shit 4.

We note that, in specific settings, a drift may be undetectable when windows are far apart from each other in the stream. However, if there are intermediate changes of concept between the distributions estimated upon the first and last windows, and with proper setting of the observation windows, it may be possible to trace the evolution of the drift over time and detect it without true labels (Dyer et al. 2014; Souza et al. 2015a, b). Figure 4 illustrates such a case.

Kull and Flach (2014) extend the work of Moreno-Torres et al. (2012) by introducing graphical notations of the dataset shifts types mentioned above, and 12 new additional sub-types of shifts. We point the interested reader to this paper for further information on this topic. Oppositely, Kelly et al. (1999) and Tsymbal (2004) offer a simplified view that is often enough to specify a concept drift problem. According to them, *concept drift* occurs when $P(X)$, $P(X|Y)$ or $P(Y|X)$ change.
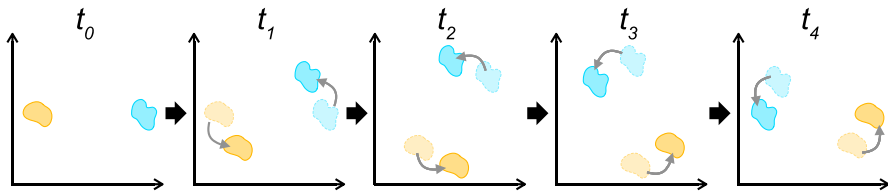
Fig. 4 Illustration of a case of traceable incremental drift. There are two classes, distinguishable by their unique color. This illustration presents five snapshots that represent the evolution of the data over time. The thick arrow represents a passage from one snapshot to the following. The curved arrow represents the movement of the class in the feature space. A brighter class with a dashed border represents the previous position. Notice that if we only compare t_0 with t_4, we have a case that approximates $P_A(Y|X) \neq P_B(Y|X)$ and $P_A(X) = P_B(X)$, which is undetectable without true labels. However, with the support of the intermediate distributions, we can trace the geometric evolution of the data and therefore distinguish both distributions
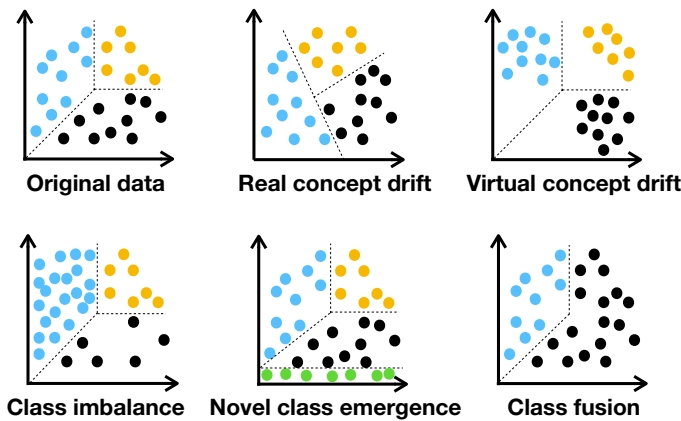

Fig. 5 Some types of concept drifts in data stream frequently found in literature. Dashed lines indicate the separation margin between the classes

The cases where $P(X)$ changes, while $P(Y|X)$ does not change, are referred to as *virtual drift*. Opposite cases, where $P(Y|X)$ changes while $P(X)$ does not, occur due to alterations in the *hidden context*. Hidden context is the information that is not included into the observable predictive features but is relevant to determine the class label (Harries et al. 1998). Furthermore, severe changes in $P(Y)$ can lead to class imbalance. Such changes in the class distribution can make a majority class to become a minority class and vice-versa in the course of a stream (Maletzke et al. 2018).

Perceiving the occurrence of a concept drift can carry different meanings and consequences depending on the application. For instance, in an application where there is interest in detecting new classes of data, a change where a new cluster of data points appears may represent the emergence of a novelty. Figure 5 illustrates different practical types of drift found in data stream literature. In this example, circles represent instances, and colors represent classes. The figure also shows the decision boundary that discriminates the classes.

2.5 Data stream realization

One point of concern is the nature of the sequence and how it translates to the divergences in how consecutive instances are observed.

In certain sequences, the last observed instance is a transformation of previous instances and therefore *could not* have existed before their materialization (e.g., Harries 1999; Zhu 2010). This is the case of most data from time-series problems. In a considerable amount of them, the observed data are complete, *i.e.*, we observe all instances of the problem, and the practical objective is to predict future readings according to the data trend. In that sense, each individual observation can be considered of little importance. We highlight that in those cases, the feature-values registered for each observation are highly dependent on previous observations. For that reason, these data are the most affected by temporal dependence. We name sequences under this scenario *materialization sequences*.

We recall that instances are distributed according to a background concept that evolves over time. However, in certain cases, the order in which instances were actually observed does not imply the necessary order of their materialization, that is, the order in which the instances started to exist (e.g., Zliobaite 2011; Ikonomovska et al. 2011; Katakis et al. 2009; Souza et al. 2015b; Reis et al. 2016). One example is the data collected by a mosquito trap that measures flight characteristics of insects using sensors. While multiple insects may coexist in the trap's vicinity at the same time, the ones that fly into the trap do so in a somewhat randomized order. However, all those insects have their behaviors affected by shared environmental factors, which change over time. In most cases that follow this structure, the observed data are only a sample of a more extensive set of instances that may not ever be observed, and the practical objective is to determine the class of each instance. In that sense, each observation is considered of high importance. We name sequences under the described setting *observational sequences*.

The arrangement of data points in a sequence is commonly tied to time, be it the order in which data points were observed or materialized in the world. We call sequences that have their arrangement tied to time *temporal sequences*. However, not all streaming data are tied to the chronological order of events.

Therefore, another relevant aspect of data stream is the physical nature of the sequence. Frequently, a stream is not chronological even though there is a logical sequence (e.g., Blackard and Dean 1999). One example is the analysis of the pavement quality of a road (e.g., Souza 2018; Souza et al. 2018). The extent of the road can be split into sections that are analyzed individually, and the order of such sections can follow their spatial positions. Therefore, the resulting sequence follows a logical sequence, yet the actual time when data for each section were collected is irrelevant and interchangeable. When the order of the instances is related to their spatial disposition, we call the resulting sequence *spatial sequence*. Finally, sequences not tied to time nor space are called *logical sequences*.

When the concept behind the instances in a sequence is tied to either time or space, a relevant aspect is the spacing between instances. In typical materialization and spatial sequences, instances can generally be observed at regular time/space inter-
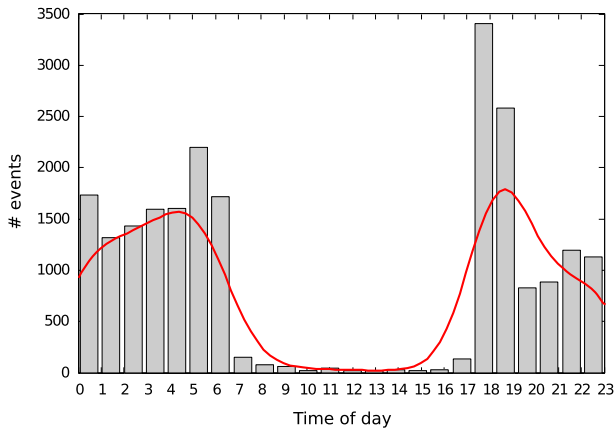
Fig. 6 Circadian rythm of *Culex quinquefasciatus* (Souza 2016). Each bar represent the amount of insect passages over the trap's sensor given a time of day

vals. However, there are cases where the time/space between consecutive observations vary, and this setting poses particular complications for observational sequences. For example, in the mosquito trap application mentioned above, we know that different species of flying insects show more or less activity according to their circadian rhythm (Shinkawa et al. 1994). Figure 6 illustrates the circadian rhythm of *Culex quinquefasciatus* mosquitoes, measured by the trap's sensor over a week.

We can see that mosquitoes of this species abruptly become inactive in the dawn and resume their activity in the dusk. Observations done by the trap rely on the activity of insects so that it is improbable to collect data while specimens are inactive. However, even if the trap is not making observations, time passes and environmental condition changes, and therefore the flight characteristics of insects also changes. If the sequence does not have timestamps for each observation, we could probably observe an abrupt and inexplicable change of behavior for *Culex quinquefasciatus*. On the other hand, with timestamps, we can notice that we lacked data for a prolonged period in which the behavior might have changed.

The example mentioned above illustrates the importance of temporal or spatial marks to understand concept drifts. In similar cases where the order of observations defines the sequence of the stream according to when they were made, if the observations are not evenly distributed and are tied to the temporal or spatial progression of the background environment, it is essential to include timestamps or longitudinal data to understand changes.

For the sake of completeness, there are cases where data are considered to be a stream only due to its long length, although it lacks logical ordering (e.g., Cattral et al. 2002; Vergara et al. 2012). If an instance is as likely to be observed at time t as at time t', $t \neq t'$, $\forall t, t'$, then there is no concept drift. In fact, any two windows of data are going to be equally distributed, since both are uniform samples of the data. In such cases, forgetting mechanisms to discard old data are not beneficial, but should not be unfavorable to the performance of the classifier either. We call sequences under this setting *unordered sequences* (e.g., Cattral et al. 2002).

2.6 Stream classification

Classification is probably the most common task in data mining and a topic of active research in data stream (Street and Kim 2001; Bifet et al. 2013; Souza et al. 2015b; Gomes et al. 2017). Classification is the process of inducing a general model from known data (training data), and then using such a model to predict class labels (discrete values) for previously unknown data objects (test data). Differently from batch learning, in data stream or online learning, the test examples arrive continuously in an orderly fashion over time, and a classifier should generally predict the label of each instance in real-time or at least before the arrival of the next example.

In classification, the objective is to build a model that approximates the true relation between instances ($\mathcal{X}$) and their respective class-labels ($\mathcal{Y}$), and, therefore, is capable of predicting the class-labels of unlabeled instances. In other words, we want to build $h : \mathcal{X} \rightarrow \mathcal{Y}$ such that $h(\mathbf{X}_t) = y_t$. The task of inducing h from labeled instances is referred to as *training*.

In batch learning, the classification model h is typically induced beforehand using a training set of labeled instances. However, in data stream classification, several approaches differ significantly regarding which data are used for training.

A recent approach assumes that labeled data for all possible concepts are available before the stream begins to be processed (Reis et al. 2018a, b). In that case, individual classifiers are trained for each concept without regard for the data stream at first. Such models are later *deployed* to classify examples in the stream. However, there is not a single attribute that can easily identify, which is the concept of current data. Therefore, we need means to detect which is the adequate classifier to be used for recent data.

Nevertheless, the most common setting in the data stream community is that there is no separate training set to induce a definitive model h. Thus, the model needs to be constructed and updated on the fly as new data are observed (Zliobaite et al. 2015). Among approaches in this setting, some methods incrementally evolve the classification model but do not adapt to changes in data. Such methods are capable of dealing with the fast pace of streaming data and keep low usage of memory. One example is the Very Fast Decision Tree—VFDT (Domingos and Hulten 2000), where new incoming examples update the statistics of the leaves of a tree-based model. As more instances are observed, the statistics are used to decide which and when leaves are split into new leaves, resulting in the growth of the tree. Recently, VFDT was adapted as the newly introduced Extremely Fast Decision Tree to become faster (Manapragada et al. 2018).

However, due to changes in data distribution, the learner must incrementally adapt its model h over time or performs updates when necessary to maintain stable predictive performance. Therefore, we also have a sequence of models $h_1, \ldots, h_i$, which can be discarded or reused in recurring situations. There are two main approaches to deal with concept drifts in classification problems (Khamassi et al. 2018): (i) evolving models, and (ii) adaptive models. Gama et al. (2014) names such approaches of (*i*) blind, and (*ii*) informed.

The evolving models update the learner at regular intervals without considering whether changes have occurred. To do so, the model uses mechanisms for learning

new concepts and forgetting the old ones. A common approach is a sliding window with a fixed or variable length to store training examples or by weighting the data by age/utility (Klinkenberg 2004). Such strategies consider that the most recent data are more representative of the current concept, so we can discard old data or assign them less weight. The main weakness of this approach is that the forgetting of old concepts is carried out at a constant speed for the whole time. Therefore, old data are discarded even when changes are not happening.

The evolving methods are naturally able to handle gradual and incremental drifts. Examples of work that implements evolving approaches are the algorithms from the FLORA family (Widmer and Kubat 1996). In FLORA2, incoming examples are added to the window, and the oldest ones are deleted. A naive approach that falls into the same category is periodically retraining a new classification model with the last observed instances. Besides, some algorithms recently proposed, such as COMPOSE (Dyer et al. 2014) and SCARGC (Souza et al. 2015b), use the sliding window strategy to deal with incremental changes in scenarios in which the actual labels of test instances are never available to the learner.

Adaptive models explicitly detect concept changes using drift detectors, updating the model only when changes are flagged. One of the advantages of explicit detection is the production of information about the dynamics of the data generation process and the reduced amount of updates in conditions without concept changes. An example of relevant work that uses drift detection is the extension of VFDT called Concept-adapting Very Fast Decision Trees—CVFDT (Hulten et al. 2001). CVFDT works by keeping its model consistent with a sliding window of examples. However, it does not need to learn a new model from scratch every time a new example arrives; instead, it updates the sufficient statistics at its nodes by incrementing the counts corresponding to the new examples, and decrementing the counts corresponding to the oldest example in the window (which now needs to be forgotten). If the concept is changing, some splits that previously passed the Hoeffding test will no longer do so, because an alternative attribute now has higher gain. In this case, CVFDT begins to grow an alternative subtree with the new best attribute at its root. When this alternate subtree becomes more accurate on new data than the old one, the old subtree is replaced by the new one.

According to Khamassi et al. (2018), the main issues of the approaches that uses drift detectors are related to knowing (*i*) how to track concept drift, (*ii*) which data to keep and which data to forget, and (*iii*) how to adapt the learner parameters and structure to react according to the requirements of the new environment.

2.7 Drift detection

Drift or change detection is the task of identifying significant data distribution changes in a stream. Although drift detection is a common mechanism of adaptive stream classifiers as a trigger for model updates, it is also a separated task from the classification process that contributes to other real applications as those related to signal analysis or time-series.

For example, change detection can be used to provide alerts when the value of a stock is falling in the market (Oh and Kim 2002) or identifying a fault in the monitoring of industrial processes (Venkatasubramanian et al. 2003). An important application of change detection methods is in burst detection (Gama 2010). Burst regions are time intervals in which some feature values are unexpected. For example, gamma-ray burst in astronomical data might be associated with the death of massive stars; bursts in document streams might be valid indicators of emerging topics, and so on.

In classification problems, drift detection methods are categorized into two major groups according to the availability of labeled data in the stream (Faithfull et al. 2019): (*i*) supervised, and (*ii*) unsupervised. Supervised drift detection methods assume the immediate availability of class labels of each instance. These methods surveil indicators of classification performance, such as accuracy to detect drifts. On the other hand, when the class labels are delayed or unavailable, unsupervised methods detect drifts by comparing data distributions at different time intervals.

Based on the taxonomy proposed by Gama et al. (2014) of dimensions which characterize drift detection methods, we consider three main categories:

(I) Methods based on differences between two distributions. In this approach, the methods monitor the distributions of two data windows. We can consider a reference window with old data and a detection window composed of recent data. These windows are compared using statistical tests, with the null hypothesis that the data of both windows are drawn from the same distribution. Thus, a concept drift is flagged when the test rejects the null hypothesis. The windows can contain unsupervised information as the raw data, learner parameters, classifier's outputs such as probabilities estimate or classification scores, as well as supervised information such as the error rate of the classifier or even the class labels.

Some parameters are fundamental to the success of these methods such as how to measure the change and how to determine the size of the windows. To measure the change, different types of statistical tests as univariate or multivariate and parametric or non-parametric can be employed. Examples of tests are the Kullback–Leibler divergence (Dasu et al. 2006), Hotelling's t^2 (Hotelling 1992), semi-parametric log-likelihood—SPLL (Kuncheva 2013), Kolmogorov–Smirnov (Reis et al. 2016). Regarding the window size, it is important to note that a window smaller than the changing rate may lead to a higher number of false negative detections, and a window larger than the changing rate may delay the detections. Most of the work done is based on fixed-size windows, where delay in detections is frequent (Ganti et al. 1999; Kifer et al. 2004; Dasu et al. 2006). Other pieces of work consider windows with dynamic size. For example, ADWIN (Bifet and Gavalda 2007) finds two windows of different sizes through multiple tests between consecutive sub-windows within a window with fixed and large enough size. To detect drifts, ADWIN uses the Hoeffding bound to compare the sub-windows. Similarly to ADWIN, SEED (Huang et al. 2014) uses two sub-windows with dynamic sizes but also performs block compressions to reduce the number of window comparisons. It also computes the volatility shift to describe the relationship of proximity between consecutive drift points in the stream.

(II) Methods based on sequential analysis. The method Sequential Probability Ratio Test—SPRT (Wald 1947) is the foundation of detection methods such as CUSUM and Page–Hinkley (Page 1954). To better understand SPRT, consider a sub-

sequence of N examples from the stream $\mathcal{DS} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N, \ldots\}$ where the subset of instances $\mathcal{D}_1 = \{\mathbf{X}_1, \ldots, \mathbf{X}_w\}$ with $1 < w < N$ is generated from an unknown distribution P_A and the subset $\mathcal{D}_2 = \{\mathbf{X}_{w+1}, \ldots, \mathbf{X}_N\}$ is generated from another unknown distribution P_B. A change is declared at time w if the probability of observing examples under the distribution P_B is significantly higher than the P_A. For this verification, SPRT tests the logarithm of the likelihood ratio considering the two distributions. The main difference compared with the approaches previously discussed is that the test of SPRT is made sequentially one by one with different values of w, until the decision to accept or refuse the null hypothesis that P_A and P_B are the same distribution. SPRT is a classic method proposed in statistics, and data stream applications still employ it to detect changes with competitive performance (Faithfull et al. 2019).

(III) Methods based on statistical process control For decades, the quality control of products in continuous manufacturing is made using standard statistical techniques called control charts. Different methods such as DDM (Gama et al. 2004), EDDM (Baena-Garcia et al. 2006), and EWMA (Ross et al. 2012) are based on these statistical techniques to detect changes in data stream. These drift detection methods consider the classification problem as a statistical process and monitor the evolution of some performance indicators, such as the error rate, to apply heuristics to find points of change. For example, the method DDM considers three different states for the classification error evolution: (*i*) *in-control*, when the error is stable; (*ii*) *out-of-control*, when the error is increased significantly as compared to the recent past; and (*iii*) *warning*, when the error is increasing but has not reached the out-of-control state. The method stores the data in a short-term memory during the *warning* state and rebuild the classification model with this data when the error state is changed to *out-of-control*.

The method employs a set of rules considering the mean and variance of the Binomial distribution of the classifier's errors to define the threshold of the states. An advantage of this method is that the rate of a change can be measured according to the number of examples or the time between the *warning* and *out-of-control* states. In this case, short times indicate fast changes, while longer times indicate slower changes. Inspired by DDM, the EDDM also takes into account the distance between consecutive errors as opposed to considering only the error magnitude.

The work of Gama et al. (2014), Ditzler et al. (2015), and Khamassi et al. (2018) provide interesting reviews about different drift detection methods from the literature. Also, Gonçalves Jr et al. (2014) performs an experimental comparison of drift detection methods.

3 Stream datasets from literature

Bifet et al. (2009) note the difficulty of finding large real-world datasets for public benchmarking, especially with substantial concept drift. We would like to quantify how this difficulty impacts the variety of datasets used in data stream research. Therefore, we performed a broad literature review over the last two decades. We analyzed more than 150 papers from high-impact conferences and top-tier journals to find the most used datasets. Unlike batch learning, in which a few hundred static datasets are

available for evaluation, the data stream learning community has supported their findings in approximately 15 real-world datasets. In what follows, we describe the most popular datasets.

Electricity (Harries 1999). This dataset probably is the most used for the tasks of stream classification and drift detection. The data are from the Australian New South Wales Electricity Market. Prices are affected by the demand and supply which are assessed every five minutes. The learning task is to predict a rise or a fall in electricity prices, given recent consumption and prices in the same and neighboring regions. The dataset contains 45,312 instances, eight attributes, and two class labels (UP and DOWN);

Forest Covertype (Blackard and Dean 1999). This dataset contains information about the forest cover type of 30×30-m cells obtained from the US Forest Service Region 2 Resource Information System. It contains 581,012 instances, 54 attributes, and seven class labels related to different forest cover types.

Poker-hand (Cattral et al. 2002). Each record of this dataset is a poker hand consisting of five playing cards drawn from a standard deck of 52. Each card is described by two attributes (suit and rank). The dataset contains 1,025,010 instances, 11 attributes, and 10 class labels related to a possible poker hand such as a *one pair, two pairs, flush, full house*, among others;

Intrusion Detection or KDDCUP99 (Tavallaee et al. 2009). This dataset is from the KDD Cup 1999 Competition. The MIT Lincoln Labs gathered such data for nine weeks. The data consist of raw TCP dump data from a local area network. The learning task is to build a predictive model capable of distinguishing between normal connections and intrusive connections such as DoS (denial-of-service), R2L (unauthorized access from a remote machine), U2R (unauthorized access to local superuser privileges), and Probing (surveillance and other probing) attacks. The original task comprises 24 training attack types. The full dataset has about five million connection records, but it is usual to consider a subset with only 10% of the size;

Airlines (Ikonomovska et al. 2011). This dataset is from the Data Expo Competition 2009. The dataset consists of flight arrival and departure records of commercial flights within the USA, from October 1987 to April 2008. The learning task is to predict whether a given flight will be delayed, given the information of the scheduled departure. The dataset contains 539,383 examples, seven attributes, and two class labels (Delayed and Not delayed);

Gas Sensor Array (Vergara et al. 2012). The dataset was gathered from January 2007 to February 2011, totaling 36 months, in a gas delivery platform facility situated at the University of California, San Diego. It comprises recordings from six distinct pure gaseous substances: Ammonia, Acetaldehyde, Acetone, Ethylene, Ethanol, and Toluene. The dataset contains 13,910 instances, where each instance consists of the measurements of 16 chemical sensors attached to an array. For each instance, *only one* of the gases is diluted in dry-air at a varying concentration at a time inside of a chamber with the sensor array. The chamber where the gases are measured avoids any interference of the dynamics of the gases to the measurements. Therefore, only the presence of the gases should induce the conductivity

of the sensors. An updated version of the dataset includes, for each instance, the concentration of the gas (Rodriguez-Lujan et al. 2014). A discrete number of concentrations was assessed for each gas. The amount and which concentrations were measured according to the gas type. *Drift* was expected in a class due to the difference in concentrations. However, the original dataset was not intended to be a streaming dataset, and each instance was sampled independently from the other ones. The dataset was originally divided into batches that do not even follow the same logical sampling order. The classification problem is to identify which gas is measured.

Luxembourg (Zliobaite 2011). This dataset was constructed using the European Social Survey 2002–2007. The task is to classify a subject concerning the internet usage as *high* or *low*. A possible source of drift is internet usage change over time. The dataset has 20 features (31 after transformation of categorical variables) based on the answers to a survey questionnaire and 1901 examples collected over five years;

Chess.com (Zliobaite 2011). This dataset comprises data from *chess.com* portal. The data consist of game records of one player over a period from December 2007 to March 2010 comprising seven attributes such as start date of the game, speed of the move in days, number of moves until the end of the game, type of the game (personal, tournament, and championship), current rating, opponent rating, and piece's color. Each player has a rating, which changes depending on achieved results. A possible source of drift is the fact that a player develops skills over time, besides engaging in different types of tournaments and competitions. The rating and the type of game determine how the system selects an opponent. The task of this data is to predict if the player will win, lose or draw a game;

Ozone (Dua and Graff 2017). This data consists of air measurements collected from 1998 to 2004 at the Houston, Galveston, and Brazoria areas. The learning task is to predict the ozone level eight hours ahead of time. The dataset has 72 attributes, 2534 examples, and two class labels (Ozone day and Normal day).

Sensor Stream (Zhu 2010). This dataset contains environmental information (temperature, humidity, light, and sensor voltage) collected from 54 sensors deployed in the Intel Berkeley Research Lab. The whole stream contains information recorded consecutively over two months (one reading every 1–3 min). The learning task is to identify the sensor ID based on the sensor data. This dataset contains 2,219,803 instances, five attributes, and 54 class labels;

Powersupply (Zhu 2010). This dataset contains hourly power supply data from an Italian electricity company. The data were collected from two sources: power supplied from the main grid and power transformed from other grids. The stream contains 3-year data from 1995 to 1998, and the learning task is to predict which hour of the day (1 out of 24 possibilities) the current power supply belongs. The argument for concept drift is that it is mainly driven by season, weather, time of the day (e.g., morning and evening), and the differences between working days and weekends. This dataset contains 29,928 instances, two attributes, and 24 class labels;

Spam Assassin Corpus (Katakis et al. 2009). This dataset consists of email messages chronologically ordered according to their date and time of arrival. The

learning task is to identify if an email contains spam or a legitimate message. In this problem, the authors consider the occurrence of abrupt and gradual drifts. For the first case, consider that the user can inform the machine learning system of email filtering about his/her interests by marking messages as "interesting" or "junk". For example, a user subscribed to a mailing list might suddenly stop to be interested in messages containing smartphone reviews just after the purchase of a device. A situation where both abrupt and gradual concept drifts can occur is the user regaining interest in topics that he has been previously interested in. The dataset has 9324 examples and 97,851 attributes. There are two classes, legitimate and spam, with the ratio around 25% of spam;

Rialto Bridge Timelapse (Losing et al. 2016). This dataset was built using images extracted from time-lapse videos captured by a webcam with a fixed position. The recordings cover 20 consecutive days during May–June 2016, capturing ten colorful buildings next to the famous Rialto bridge in Venice. Each captured image was segmented to cover each building and generating ten different instances. The classification problem of this dataset is to identify the correct building. Continuously changing weather and lighting conditions affect the data representation over time. Each one of the ten classes of this dataset has 8225 examples encoded in a normalized 27-dimensional RGB histogram, totaling 82,250 examples;

Outdoor Objects (Losing et al. 2015). This dataset was built from images recorded by a smartphone camera in a garden environment. The task is to classify 40 different objects such as balls, shoes, pliers, cans, among others. One hundred pictures were taken of each object under varying lighting conditions (sunny and cloudy), affecting the color-based representation, and from different distances and positions. Altogether 4000 images were recorded and arranged in a temporal order. The examples from this dataset are represented using a normalized 21-dimensional RG-Chromaticity histogram;

Keystroke (Souza et al. 2015b). It is a subset of the larger CMU dataset (Killourhy and Maxion 2010), where 51 users type the password ".tie5Roanl" plus the *Enter* key 400 times captured in eight sessions performed in different days. In the Keystroke data, the typing rhythm is used to recognize four different users. In this classification task, ten features are extracted from the flight time for each pressed key. The flight time is the time difference between the instants when a key is released, and the next key is pressed. This dataset contains 1600 instances that incrementally evolve due to the users' practice;

NOAA Weather (Ditzler and Polikar 2013). The dataset consists of weather measurements collected over 50 years at Bellevue, Nebraska by the National Oceanic and Atmospheric Administration (NOAA). This dataset contains eight features: temperature, dew point, sea-level pressure, visibility, average wind speed, max sustained wind-speed, minimum temperature, and maximum temperature. The learning task is to determine whether it will rain or not. The dataset contains 18,159 daily readings of which 5698 are *rain* and the remaining 12,461 are *no rain*.

Table 1 presents a summary of the characteristics of the datasets.

Table 1 Characteristics of the main stream learning datasets available for evaluation

Dataset	Instances	Features	Classes	Sequence type	Ordering
Electricity	45,312	8	2	Materialization	Temporal
Forest covertype	581,012	54	7	Observational	Spatial[a,b]
Poker-hand	1,025,010	11	10	Observational	Unordered
KDDCUP99	494,021	41	23	Observational	Temporal[b]
Airlines	539,383	7	2	Observational	Temporal
Gas sensor array	13,910	128	6	Observational	Logical[a]
Luxembourg	1901	30	2	Observational	Temporal
Chess.com	534	7	3	Observational	Temporal
Ozone	2534	72	2	Materialization	Temporal
Sensor stream	2,219,803	5	54	Materialization	Temporal
Powersupply	29,928	2	24	Materialization	Temporal
Spam assassin	9324	97,851	2	Observational	Temporal
Outdoor	4000	21	40	Observational	Temporal
Rialto	82,250	27	10	Observational	Temporal
Keystroke	1600	10	4	Observational	Temporal[b]
NOAA weather	18,159	8	2	Materialization	Temporal

[a]Details about ordering are not provided
[b]Timestamps or spatial marks are not included

The small number of real-world datasets publicly available impose restrictions on comparative studies of new proposals (Krawczyk et al. 2017). Such a lack of benchmark data leads to the use of approaches to simulate changes in static data or the generation of synthetic data with concept drifts.

Some common approaches to simulate changes in real data with static distribution are (Sobolewski and Wozniak 2013):

- *Switching the features.* To simulate concept drifts, we can switch the values of some features while maintaining the class labels of a set of data samples (Ramamurthy and Bhatnagar 2007; Zliobaite and Kuncheva 2009). For example, given a static dataset, we first split it into two samples. In the second sample, the original feature 1 replaces feature 2, the original feature 2 replaces feature 3, and so on, while the last feature substitutes feature 1. The class labels of the examples remain the same;
- *Swapping classes.* In this approach, we randomly pick two classes in the data set and swap their labels (Klinkenberg and Joachims 2000; Kuncheva and Sánchez 2008);
- *Joining classes.* We can join two or more classes in a unique class and consider this one as a new concept in the stream (Vreeken et al. 2007).
- *Reordering the data according to a hidden feature.* We can hide a feature and use it as a shared concept for the instances. In that case, we reorder the instances to group instances within the same concept together. Within the same concept, instances may be drawn uniformly to remove any other possible source of drift. If the hidden feature is nominal, the concept drifts are usually abrupt. In the case

of an ordinal hidden feature, we can simulate incremental drift. Numeric features can be turned into ordinal features so that we can more easily draw instances from the same concept (Reis et al. 2018b). This approach suits observational sequences better than materialization sequences.

Some examples of synthetic data generators widely used by the community are STAGGER (Schlimmer and Granger 1986), SEA (Street and Kim 2001), Rotating Hyperplane (Hulten et al. 2001), Random RBF (Bifet et al. 2009), LED (Breiman et al. 1984), and Waveform (Breiman et al. 1984). We can also cite the framework proposed by Narasimhamurthy and Kuncheva (2007), the Sine, Line, Plane, Circle, and Boolean datasets proposed by Minku et al. (2010), and the synthetic datasets generated by Dyer et al. (2014) and Souza et al. (2015b) to evaluate incremental changes.

The main problem of simulating drifts in real data or the use of generators is the introduction of data bias in the experimental evaluation. Data bias is the conscious or unconscious use of a particular set of data to confirm the desired finding, and that can lead to incorrect conclusions (Keogh and Kasetty 2003).

4 Criticisms to current datasets for stream learning

In addition to the reduced number of publicly available real-world stream datasets, the most used datasets have problems such as a limited number of events and a small number of concept drifts. Unfortunately, these issues can lead to biased or incorrect conclusions when assessing the performance of stream algorithms. In this section, we discuss these problems and possible consequences.

4.1 Uncertainty about changes

One of the main problems regarding the existing stream datasets is the uncertainty about the presence of concept drifts. The community frequently assumes that the performance degradation of a classifier over time is evidence of changes in the data distribution. However, concept drift is not the only cause of performance degradation, that might have other origins such as poor generalization (e.g., underfitting or overfitting (Domingos 2012)) and noisy data arriving along the stream.

Even for datasets with known presence of concept drifts, the type of change (covariate, probability, or concept shift), pattern (abrupt, gradual, incremental, or reoccurring) and the exact moment these drifts occurred are frequently unknown.

The lack of knowledge about change characteristics and when they occur, can limit the evaluation of stream algorithms. A straightforward example is the evaluation of change detection methods that use criteria such as the probability of correct change detection, the probability of false alarms, and the lag of detection (Gama et al. 2014). Due to the lack of annotation of drift location in real data, the analysis of methods such as EDDM (Baena-Garcia et al. 2006), appropriated for slow, gradual changes, and EWMA (Ross et al. 2012), fit to abrupt changes, is only possible with the aid of artificial data. Finally, the use of inappropriate datasets for the problem tackled can lead to incorrect conclusions. One example is the use of a dataset where changes

follow $P_A(Y|X) \neq P_B(Y|X)$ and $P_A(X) = P_B(X)$ to evaluate unsupervised detection algorithms.

Virtually all publications that present real data make informal assumptions regarding the existence of drift. As far as we know, Sarnelle et al. (2015) is the first to make an effort to quantify their assumptions. Although their work is limited to the settings where drift is given by the incremental and spatial displacement of the classes in the feature space, such assumptions are valid for a broader number of existing work (Dyer et al. 2014; Souza et al. 2015b). Sarnelle et al. (2015) introduce supervised means to measure the intensity of the displacement of the classes, its direction, and, more importantly, its unsupervised traceability.

Webb et al. (2018) further raise awareness of the importance of measuring drift in datasets, introducing the task of *concept drift mapping*. Particularly, they measure the divergence between consecutive snapshots (built with observation windows) of the data to represent the distributions of concepts over time. The divergence between concepts is called, in this scenario, *drift magnitude*. The magnitude of the drift can be individually measured for *marginals* ($P(X)$ and $P(Y)$) and *conditionals* ($P(X|Y)$ and $P(Y|X)$) to provide different views of the data drift, revealing more information regarding its evolution. The authors also make comparisons between drifts on specific attributes and the total drift magnitude to highlight the contribution of different attributes to the drift. Although the drift magnitude was measured with total variation distance, any dissimilarity function that applies to distributions can be employed. Finally, we note that this work intends to provide tools to describe data, rather than mechanisms to detect drift actively.

Goldenberg and Webb (2019) review many applicable dissimilarity functions to verify which are good options for measuring drift magnitude. The work targets explicitly covariate shift (changes in $P(X)$), suiting the task of unsupervised drift detection. The authors recommend Hellinger distance to measure the divergence between distributions of univariate and low-dimensional data.

When distributions are approximated by samples of numeric values, the use of Hellinger distance implies the discretization of the data with histograms. Many options have gone untested by Goldenberg and Webb (2019) and we refer the reader to González et al. (2017) for more options and to Cha and Srihari (2002) for the particularly interesting ORD, which takes the distance between different bins into account. Maletzke et al. (2019) introduces SORD, a version of ORD that exempts the discretization of numeric values to compare univariate sample distributions, and can be seen as a particular and fast-to-compute case of the Earth Mover's Distance.

Another interesting aspect to know about the data is if they contain temporal overlap, as previously discussed in Sect. 2.4 and illustrated in Fig. 2. When it is absent, one can approach the classification problem with an incremental learner that need not implement a forgetting mechanism to discard old concepts or a system to switch between previous models. This scenario is significantly less challenging than problems that must deal with temporal overlap.

We suggest a naive approach to indirectly measure temporal overlap if the concepts are known, and data from each concept can be isolated. One can build a classifier for each concept and individually test their performance on their respective concept. An additional classifier should be built with data from all concepts and tested with a test

set that also contains examples from all concepts. If the performances of the classifiers for individual concepts are, on average, superior to that of a single classifier that single-handles all concepts, we have evidence that there is temporal overlap. Otherwise, we have evidence that we do not need to use forgetting mechanisms and incremental learners.

4.2 Temporal dependence

Nearly almost ten years after the first evaluation on the Electricity data in the stream setting by Gama et al. (2004) and the use of these data by several studies (e.g., Gama et al. 2005; Baena-Garcia et al. 2006; Bifet et al. 2009, 2010b; Brzezinski and Stefanowski 2011; Chen and He 2011; Ditzler and Polikar 2013; Demsar and Bosnic 2018; Shao et al. 2018), Zliobaite (2013) pointed out the problems of this dataset related to the temporal dependence of class labels.

Suppose we employ a naive classifier that predicts the next label to be the same as the current label. This classifier will be our baseline henceforth. For instance, if the price goes UP now, the baseline will predict that the price will go UP for the next time step as well. If the labels were independent, such a predictor would achieve 51% given the class proportions of this particular dataset. However, if we test such an approach on the Electricity dataset as it is, we obtain a much higher accuracy of 85%. Therefore, the labels are not independent, since there are long periods of consecutive UP and long periods of consecutive DOWN labels.

Zliobaite (2013) discusses the problem of temporal dependence for the Electricity dataset; however, another two popular datasets, Forest Covertype, and Poker-hand, also have the same problem. For example, in the Forest Covertype the data are probably organized according to the geographical location of the observations, although the dataset does not include annotations for the position. Thus, there is a high probability that neighboring regions are of the same class. In the Pokerhand, we have a more significant issue. The MOA's website provides a supposedly normalized version of the dataset and a link for the original version at UCI Repository. The issue is that, besides not really having been normalized, MOA's version has a different ordering for the instances. While the original version does not present temporal dependence and the No-Change baseline achieves 43% accuracy (which is the expected accuracy if there is no temporal dependence, given the proportion of the classes on this particular dataset), the same baseline achieves staggering 75% accuracy on MOA's normalized version. We can only wonder whether this temporal dependence was purposefully implanted into the data and why. From now on, we will consider the MOA's version to illustrate the effects of temporal dependence better.

Figure 7 presents the prequential accuracy of two classifiers: *i*) Naive Bayes with Drift Detection Method (DDM) (Gama et al. 2004), and *ii*) the baseline classifier No-Change that predicts the current label to the next event for the three mentioned datasets. Given a sequential dataset, in the prequential procedure (or test-then-train), every example is first used for testing and then for updating the model.

In all cases, the baseline classifier surpasses the results of the classifier that detects changes and periodically updates their model. For Electricity data, the No-Change clas-
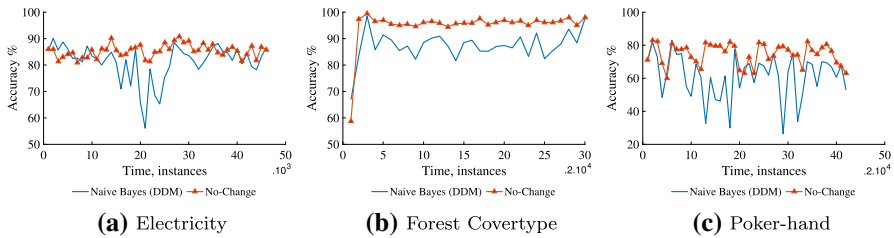
Fig. 7 Prequential accuracy of Naive Bayes with Drift Detection Method (DDM) and the baseline classifier No-Change on the Electricity, Forest Covertype, and Poker-hand datasets

sifier shows an accuracy of 85.33%, while the Naive Bayes with DDM achieves only 81.23%. For Forest Covertype, the baseline presents 95.07% of accuracy, and Naive Bayes with DDM has 88.04%. In the Poker-hand dataset, the No-Change presents an accuracy of 74.51%, and the Naive Bayes with DDM achieves just 61.96%.

In this sense, a new proposal that uses a solution based on the temporal dependence of the examples probably will show promising results on these data. However, such a good performance does not necessarily mean that the classifier has a good generalization power or it adapts well to changes.

Bifet et al. (2013) proposed a new evaluation measure (Kappa-Temporal) to avoid biased conclusions. Kappa-Temporal considers the difference between the prequential accuracy of a given classifier and the accuracy achieved by the naive classifier that ever predicts the last seen class label. However, we argue that this measure is a palliative solution to be used in the evaluation process of stream learning methods to mitigate the consequences generated by a characteristic inherent to some datasets. Further, we add that the baseline No-Change and the measure Kappa-Temporal are not well suited to compare with and evaluate classifiers that do not rely on true labels, like those that make use of unsupervised drift detection methods, since the baseline and the measure depend on such unavailable piece of information.

4.3 Data bias

Due to the reduced number of real datasets, we frequently come across stream evaluations that consider three, two, or even only one real dataset together with a larger number of synthetic data. The main problem of this practice is the possibility of data bias, as previously discussed in Sect. 3.

With a reduced number of datasets, we can demonstrate any findings we wish (Keogh and Kasetty 2003). For example, consider the use of three datasets to compare the classification performance of the Naive Bayes algorithm with two different drift detectors: DDM and CUSUM (Alippi and Roveri 2008).

In the first scenario (Fig. 8a), if we consider the Forest Covertype, Gas Sensor Array, and Ozone datasets, our obtained results would suggest that DDM outperforms CUSUM. However, if we consider a second scenario with the datasets NOAA, KDD-Cup99, and Luxembourg, we would conclude that both methods perform very similarly (Fig. 8b). On the other hand, in a third scenario where we chose the datasets Sensor
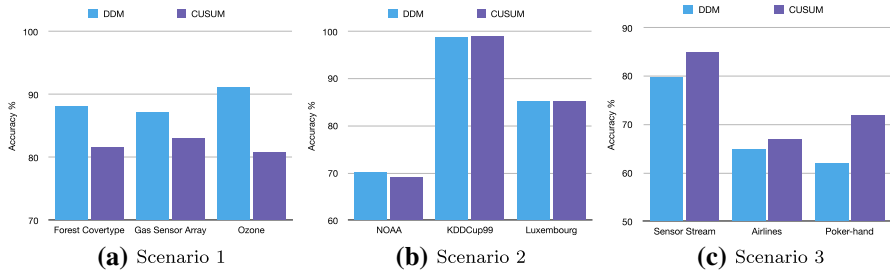
Fig. 8 In the first scenario (**a**), we consider Forest Covertype, Gas Sensor Array, and Ozone. In the second scenario (**b**), we consider NOAA Weather, KDDCup99, and Luxembourg. In the third scenario (**c**), we consider Sensor Stream, Airlines, and Poker-hand. According to the evaluated datasets, our conclusions can may be biased

Stream, Airlines, and Poker-hand, we can conclude that the CUSUM outperforms DDM (Fig. 8c).

The results presented in Fig. 8, allow us to state that the use of a reduced amount of datasets and the "right" choice of them can lead to biased conclusions. To avoid this problem, we claim by the need of a sufficiently large benchmarking data that covers different properties for stream learning, as it is already usual in batch learning.

Besides the reduced number of datasets, the typical procedure employed for evaluating the performance of adaptive learning models could also be responsible for leading to biased conclusions. As noted by Zliobaite (2014), the standard procedure, named Prequential (or test-then-train) (Gama et al. 2013), allows processing a dataset only once in the fixed sequential order. The positions where and how changes happen remain fixed; thus, a single test concludes how well a model would adapt to this fixed configuration of changes. While different learning models have different adaptation rates, the results on a fixed test snapshot with a few changes may not be sufficient to generalize how this adaptive model would perform online on a given problem. To make the evaluation more confident, Zliobaite (2014) proposes the employment of multiple tests with variations of the original dataset. The various tests are generated by permuting the data order in a controlled way to preserve local distributions.

4.4 Insufficient amount of instances

Stream predictive models that operate in changing environments have different requirements from the traditional batch setting. The three main requirements are (Gama et al. 2014):

1. Detect concept drifts (and adapt if needed) as soon as possible;
2. Distinguish drifts from noisy data;
3. Operate faster than the example arrival time and use a fixed amount of memory for any storage.

Here, we call attention to the third requirement. As data stream is frequently defined as an infinite sequence of examples, accommodating such data in the machine's main

memory is considered impractical or infeasible. However, this definition is inconsistent with the number of instances present in the commonly used stream datasets.

From the 16 popular stream datasets presented in Table 1, only two of them have more than one million examples (Poker-hand and Sensor Stream). Further, more than half have less than 50,000 examples, an amount that can be handled by most of batch learning algorithms. In general, these numbers of examples do not represent a challenge to data processing and storage. One possibility is that researchers in the community might feel challenged enough do design memory-efficient algorithms. In reality, we have noticed that very few papers analyze the memory requirements, with some exceptions, such as the system streamDM-C++ (Bifet et al. 2017).

4.5 Lack of complex distributions

For many real-world applications such as financial fraud detection, natural disaster prediction, spam filter, fault monitoring, or disease diagnosis, we have an interest in events that occur with a very low frequency. In these cases, some classes are difficult or expensive to collect. Consequently, the classes are not equally represented in the data which leads to the problem of *class imbalance* or *skewed class distributions* (Batista et al. 2004). Class imbalance can cause a bias towards the majority class, and the classifiers may tend to misclassify minority class examples due to the poor generalization (Wang et al. 2013).

The machine learning community has widely researched the class imbalance problem for more than 20 years (Chawla et al. 2004). However, this issue is still challenging and focus of intensive research in the static learning setup (Yang and Wu 2006). Although class imbalance and concept drift are intimately related when changes occur in prior probabilities $P(Y)$, learning with class imbalance has still received little attention on stream learning (Hoens et al. 2012; Ghazikhani et al. 2013; Krawczyk et al. 2017). As recently noted by Wang et al. (2018), most existing work in stream learning focuses on the concept drift in posterior probabilities (i.e., real concept drift or changes in $P(Y|X)$, as discussed in Sect. 2.2) and most proposed concept drift detection approaches are designed for and tested on balanced data streams.

Differently from static learning, in data stream setting the class distribution is not fixed. Instead, the class ratio varies, and the relationship between majority and minority classes may change over time. It becomes even more complicated in multi-class problems.

We believe that the lack of real stream datasets with complex distributions that contain changes in both $P(Y)$ and $P(Y|X)$ (or $P(Y)$ and $P(X)$) limit the research and evaluation of data stream research in realistic scenarios. For example, Fig. 9a illustrates the changes in the classes proportion in the Electricity dataset given a window with an arbitrary size of 1000 instances. The class distribution barely changes over time. As this data does not contain class imbalance, an alternative is to under-sample one of the classes as proposed in the evaluation of Learn[++].NIE algorithm (Ditzler and Polikar 2013). However, this practice can be interpreted as a modification in the real problem characteristics.
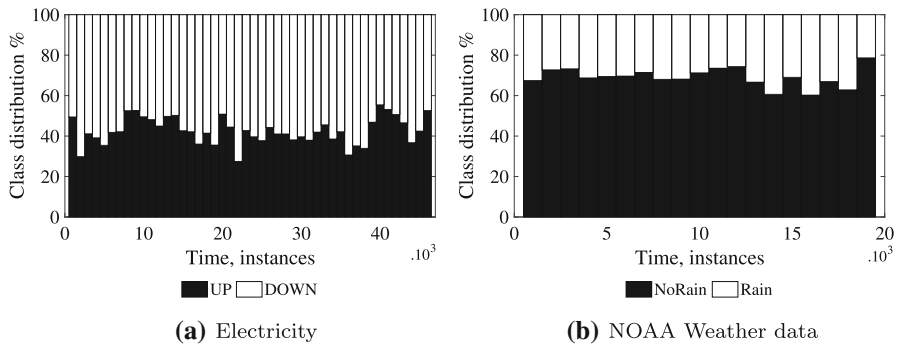
Fig. 9 Changes in data distribution over time for the classes UP/DOWN from Electricity data and Rain/NoRain from NOAA Weather data. Each bar represents the counting of 1000 consecutive examples from the stream

In recent work, Wang et al. (2018) proposes the use of PAKDD 2009 credit card (Linhart et al. 2009), UDI Twitter Crawl (Li et al. 2012), and NOAA Weather as real-world datasets to evaluate different approaches for imbalanced class distributions on stream learning.

The PAKDD data were collected from the private label credit card operation of a Brazilian retail chain. The task of this problem is to identify whether the client has a good or bad credit, where the "bad" credit is the minority class with 9,868 examples taking 19.75% of the 49,973 examples. This dataset has gradual changes since a client with bad credit can improve their status by meeting their financial commitments over time. In the same way, a good client can stop paying their debts, changing the status to bad. In the UDI Twitter Crawl, the task is to predict the tweet topic. To build this imbalanced stream dataset was chosen a subset of 8774 examples from the original data that include 50 million tweets posted from 2008 to 2011. Next, the problem was reduced for two classes by using only two out of seven possible topics. As noted by the authors, the tweet topic change can be much faster and more noticeable when compared to PAKDD 2009 data. Both datasets are important contributions. However, they present some of the previously discussed drawbacks, such as uncertainty about changes and an insufficient amount of examples. For the last data evaluated, NOAA Weather, although the majority class has 12,461 examples (68.62%), this ratio is almost constant over time, as shown in Fig. 9b. Thus, there is a need for representative stream data with more complex distribution changes over time to evaluate the problem of imbalanced classes better.

4.6 Streaming as an afterthought

One glaring aspect of the data that is often used to test data stream learners is that such data are not conceptually meant to this task. The most conspicuous example is Poker-Hand dataset. We first note that the size of the dataset is small enough (around 20 megabytes) so that it can be fed to batch learners. Last, and more importantly, the nature of any variant of a Poker game inherently grants an equal chance for every

combination of cards to be drawn at any given moment. Hands drawn from a real deck of cards are independently and identically distributed so that the hands in a stream should not be bound to a background hidden, evolving concept, and there should not be temporal dependence. This means that none of the challenges that are defended to be present in streaming data actually happen in this dataset. This is reflected by the original version of Poker-hand, found at UCI Online Repository.

However, for reasons that are beyond our knowledge, the "normalized" version distributed at MOA's website has a different ordering for the hands and is biased to present temporal dependence. While this fact is not made clear on the website, we can suspect the reason is to make the dataset more challenging and interesting for benchmarking data stream algorithms, despite the ordering being unnatural. We are not against the use of reordering to repurpose a dataset to benchmark by any means. However, we call attention to two important issues: the unnatural ordering should be explicitly explained since it is the only source of streaming challenges and it is not present in the original data; and the only challenge is temporal dependence, which is still one of the least interesting problems to have in a streaming application.

Another dataset that has been repurposed for a streaming application is Gas Sensor Array. The data collection process reassured that each example is independent, involving the use of precision equipment to set the concentration of each gas before registering the measurements of the sensor array. Only a discrete number of sparsely distributed concentrations were tested, and the different gases (that are the class labels) are never mixed together. Instead, each gas is only diluted in dry-air for the measurement. Similarly to the normalized Poker-hand, the only aspect directly associated with stream data is the unusual ordering of the examples, which is not well explained.

A less blatant example of dataset not well suited for streaming problems is Forest Covertype. The order of the examples in the dataset is likely related to their physical position in the world, which means that consecutive examples are likely to share characteristics and class labels. However, there is an immeasurable number of ways of iterating over square cells in a region, each way with its own implications in the ordering of the examples, and consequently in the temporal dependence of the stream. Yet, how the specific ordering in the dataset was achieved is unspecified, and the data do not contain the geolocation of the examples. It is also debatable if a linear representation of such data is an appropriate approach for learning tasks.

5 A real-world streaming application with concept drifts

In this paper, we introduce to the data mining community, a benchmarking dataset with different properties to evaluate stream classifiers and drift detectors. The dataset is based on a real-world streaming application based on the use of optical sensors to recognize flying insect species in real-time.

In the last years, our research group has been working in the next generation of electronic insect traps to selectively capture only certain species (Batista et al. 2011; Souza et al. 2013; Chen et al. 2014; Qi et al. 2015; Silva et al. 2015). Such smart traps use Machine Learning techniques to recognize the insects that pass in front of the sensor. The trap selectively captures species of interest such as vectors of mosquito-

borne diseases and agricultural pests, freeing all other species and, therefore, reducing the impact of this control device on the environment.

For this application, we cannot assume that a stationary stochastic process generates the data due to the existence of variations in environmental conditions that can influence the behavior of the insects. For example, temperature influences the metabolism of insects (Taylor 1963; Villarreal et al. 2017). Also, ambient conditions such as air pressure (Chadwick and Williams 1949) and humidity (Mellanby 1936) can change their flying behavior. For these reasons, the data measured by the sensor suffers from concept drifts over time, requiring adaptive models to perform the classification task of insect recognition.

We present the details of the smart trap in Sect. 5.1 and the optical sensor used into the core of the trap in Sect. 5.2. Section 5.3 details the procedures of data collection using our sensor on changing environments. Section 5.4 presents the predictive features extracted from the insect signals. Finally, we introduce the proposed stream benchmark data in Sect. 5.5.

5.1 Smart trap for insects

Controlling insect pests and vector of diseases is an important task and the main focus of the active research in the last decades. Entomologists have proposed dozens of techniques from insecticides to biological control (Medlock et al. 2012). However, these techniques can be made safer and more cost-effective with the knowledge of the spatial-temporal distributions of the insects in a certain area.

Traps are the main tool for the surveillance of insect populations. For instance, sticky traps are used in crop fields, where they are installed and collected at regular time intervals. A human expert is required to manually classify each collected individual and count the species of interest. Although sticky traps are usually inexpensive in terms of material cost, the whole procedure is expensive since it involves manual counting and classification.

The main advantage of smart traps such as proposed in our research, is their capability of counting and classifying flying insects in real-time without requesting the time and cost of analysis made by experts. Also, differently from other traps, our device deliberately does not capture non-target species, such as pollinators insects. Figure 10 illustrates a recent prototype of our device. The trap turns a fan on and off at the moment it senses a mosquito near the sensor, significantly reducing the power consumption.

In order to classify flying insects in real-time, the trap combines the optical sensor to measure the light variation over time and a circuit board to filter and record data, as well as to extract predictive features which are used by a supervised machine learning classifier.

5.2 Sensor to measure insect flying data

The proposed data in this paper was obtained from an optical sensor built with low-cost components to remotely capture information about flying insects. This sensor is the

Fig. 10 Smart Trap for counting and classifying mosquitoes in real-time using optical sensor
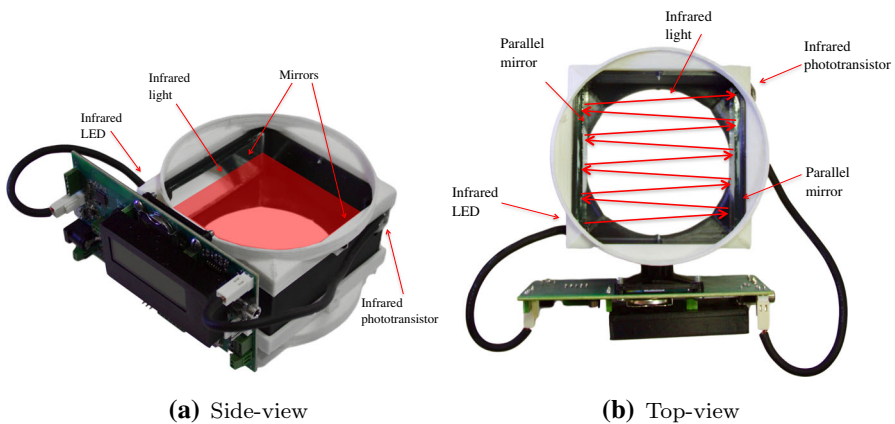

(a) Side-view   **(b)** Top-view

Fig. 11 Optical sensor to capture information about flying insects. When an insect flies across the sensor, a light variation is registered by the phototransistor as an audio signal

heart of the electronic smart trap presented in Sect. 5.1. Figure 11 shows the design of the sensor.

The sensor has two parallel mirrors face-to-face. An infrared LED uses the mirrors to create an infrared light window that is captured by a phototransistor. The infrared light bounces back and forth between the mirrors until it reaches the phototransistor. When a flying insect crosses the light, its wings and body partially occlude the light, causing small variations that are captured by the phototransistor as an audio signal. The optical device is essentially deaf to any agent that does not cross the light. This is an important feature compared to regular microphones which are susceptible to noise.

Figure 12a shows an example of data collected by the sensor given a mosquito crossing. That signal was collected from a female *Aedes aegypti* mosquito, a vector of diseases such as dengue, chikungunya, yellow, and Zika fever. The data consist of an audio fragment that usually lasts for a few tenths of a second.

(a) Signal

(b) Frequency spectrum
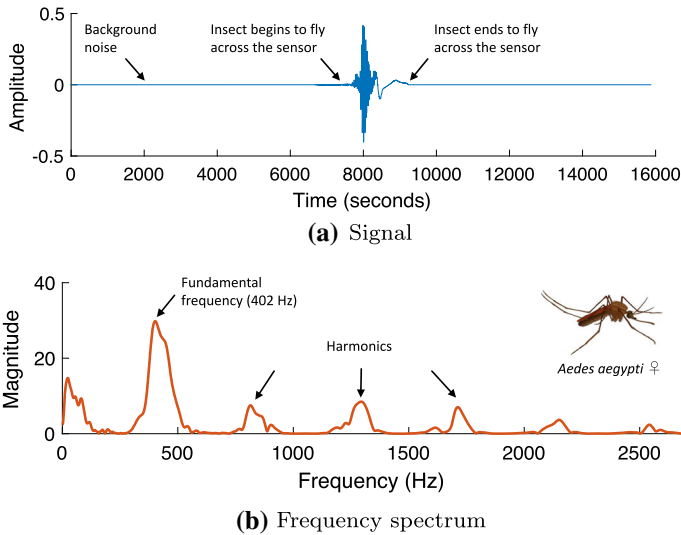
Fig. 12 A signal generated by the optical sensor given the crossing of an *Aedes aegypti* (female) through the light and the spectrum of frequencies of the same signal. From the spectrum of frequencies, we can see the wing-beat frequency of the insect (402 Hz) according to the fundamental frequency. Also, the location of harmonics in the spectrum is an important information for species discrimination

To classify the insect species, the wing-beat frequency is one of the most relevant pieces of information that can be extracted from the signals. When the signal is represented in the frequency domain, certain properties such as the fundamental frequency are made explicit. In the case of insect data, the fundamental frequency is directly related to the wing-beat frequency. Beyond the fundamental frequency, the spectrum of a signal also has harmonic components with (typically) smaller magnitudes multiples of the fundamental frequency. The position and amplitude of these harmonics also constitute important information to distinguish different insect species. Both wing-beat frequency and harmonics are shown in Fig. 12b, given the same signal generated by a female *Aedes aegypti mosquito*.

5.3 Data collection in changing environment

To build the insect stream datasets with concept drifts, we collect data from different species using our optical sensor in a non-stationary environment for three months approximately. We collected data in São Carlos, São Paulo, Brazil (University of São Paulo campus).

To know the true class label of each insect passage during data collection, we build different collector devices in which only one insect species (with many specimens) is present inside the collector. Temperature, humidity, luminosity, and air pressure sensors are positioned in the internal part of the collector.

The temperature has a direct influence on the measured data by the sensor with impact in the wing-beat frequency (Taylor 1963; Villarreal et al. 2017; Gebru et al.
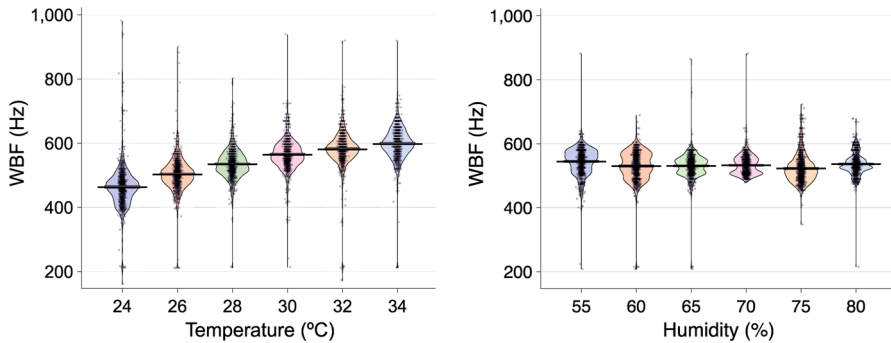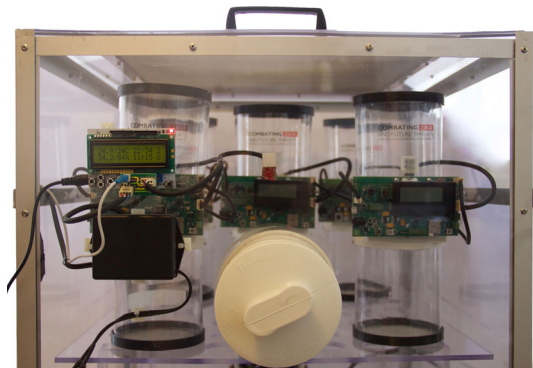
Fig. 13 Influence of temperature and humidity on the wing-beat frequency observed for *Aedes aegypti* (female) mosquitoes

Fig. 14 Chamber used to control temperature and humidity conditions in data collection


2018). However, we do not find clear evidence that humidity has any significant effect. For example, in Fig. 13 we show the WBF versus temperature and humidity for female *Aedes aegypti* mosquitoes. This plot is similar to a typical box plot, but it also shows the kernel probability density of the data at different values. To collect data for this plot, we varied temperature from 24 to 34 °C (increments of 2 °C) while keeping relative humidity constant at 70% and varied humidity from 55 to 80% (increments of 5%) while keeping the temperature constant at 28 °C.

To collect data that contemplate a wide range of environmental variation, but in a controlled manner, we built chambers where we can control temperature and humidity manually using a custom circuitry. We put the collectors inside the chambers to gather data of different species in parallel with the same environmental condition. In Fig. 14, we show a chamber with five data collectors inside.

We collected around one million instances for 17 different insect species, including mosquitoes, houseflies, bees, and wasps. For 7 of the 17 insect species, it was possible to collect the data separated by sex, totaling 21 class labels. For approximately three months, we varied the temperature from 20 to 40 °C and the humidity from 20 to 90%, considering different combinations of both variables. In Fig. 15, we show the distribution of the instances from different species over both variables. In this plot, each blue bar represents the number of insect passages given a value for humidity

Fig. 15 Number of instances observed given different values of temperature and humidity in the data collection for all species
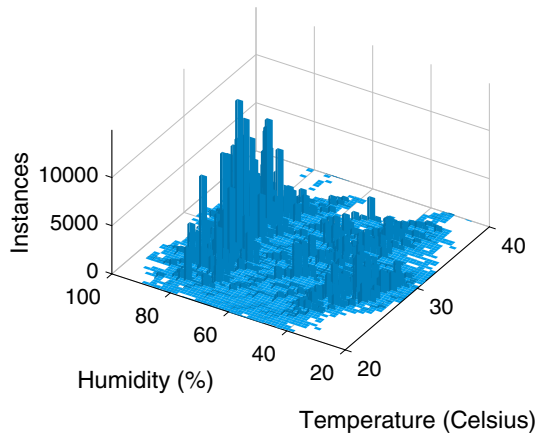
and temperature. As we can see, our data collection has contemplated a wide range of combinations, with more instances when the humidity is around 80%. We note that the proportions of observations made for different combinations of humidity and temperature do not necessarily translate to how proportionally active the insects are concerning to such variables in nature.

As some species are less active during certain times of the day or present a reduced lifetime, it was not possible to collect observations for all of them covering the entire range of variation in temperature and humidity. For these reasons, we built our datasets considering a subset with three species from both sexes, generating six class labels. We choose the following most active species:

– *Aedes aegypti.* Also known as the yellow fever mosquito, is a mosquito that can spread dengue fever, chikungunya, Zika fever, Mayaro and yellow fever viruses, and other disease agents. This mosquito originated in Africa (Mousson et al. 2005), but is now found in tropical, subtropical and temperate regions throughout the world (Eisen and Moore 2013);

– *Aedes albopictus.* Also known as Asian tiger mosquito or forest mosquito, is an invasive species that can be currently found in temperate and tropical Asia (its area of origin), Europe, North and South America, Africa and a number of locations in the Pacific and Indian Oceans (Paupy et al. 2009). It is an epidemiologically important vector for the transmission of many viral pathogens, including yellow fever, dengue fever, and Chikungunya fever, as well as several filarial nematodes such as *Dirofilaria immitis* (Gratz 2004);

– *Culex quinquefasciatus.* Commonly known as the southern house mosquito, is a medium-sized mosquito found in tropical and subtropical regions of the world. It is the vector of Wuchereria bancrofti, avian malaria, and arboviruses including St. Louis encephalitis virus, Western equine encephalitis virus, and West Nile virus (Bartholomay et al. 2010).

The anatomy of the three species is very similar, especially when we consider species from the same genus (*Aedes aegypti* and *Aedes albopictus*). This similarity is
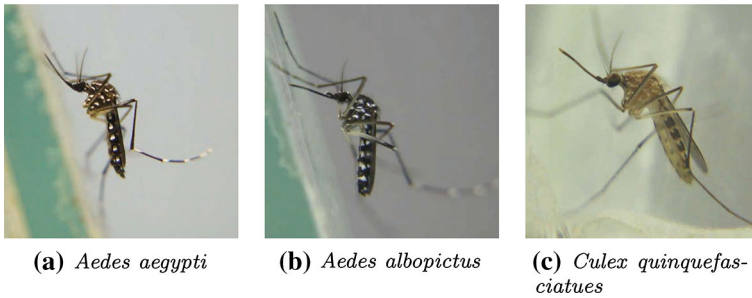
(a) *Aedes aegypti*   **(b)** *Aedes albopictus*   **(c)** *Culex quinquefas-
ciatues*

Fig. 16 Adult male mosquitoes from the species present in our datasets. All photographs are made by
Michele M. Cutwa (Cutwa and O'Meara 2006)

also observed in the flight of the insects and consequently in the measured data by the
sensor. Figure 16 illustrates with photos the species present in our datasets.

5.4 Features extraction

As we consider a more substantial number of species, it is clear from the pigeonhole
principle (Ajtai 1988) that to classify those species with high accuracy it is required
additional features than only the wing-beat frequency (WBF). For instance, Fig. 17
illustrates the distributions of the wing beat frequency for 15 species across all tempera-
tures for which we possess data. The wing-beat frequency is one of the most distinctive
attributes available for this application. In this figure, we can see that although some
species show a well-defined peak in the mean values of WBF, there is a significant
overlap among the species. In this sense, the use of only this feature to classify the
insect species can be inaccurate. We reiterate that, while this illustration can accurately
depict the range of frequencies reached by each species, the shapes of the distribu-
tions are likely skewed. This distortion is due to data collection bias: the number of
observations done for each temperature varied, and the proportions of observations
across different temperatures do not necessarily reflect a pattern of activity for insects
in nature.

Our signal, although optical, is very similar to audio, as previously shown in Fig. 12,
and consequently high-dimensional. Thus, we employ signal processing techniques
to extract additional discriminative features from data.

Each audio file was pre-processed and transformed in to a feature vector. We
extracted a series of features such as the wing-beat frequency, complexity measures
of the signal spectrum, statistics from temporal representation, among others. For the
benchmarking data, we provide 33 features related to the energy sum of frequency
peaks and harmonics positions.

5.5 Insect stream data

Given the impact of temperature on the measured data by the optical sensor leading
to the occurrence of concept drifts, we built our benchmarking data based on changes
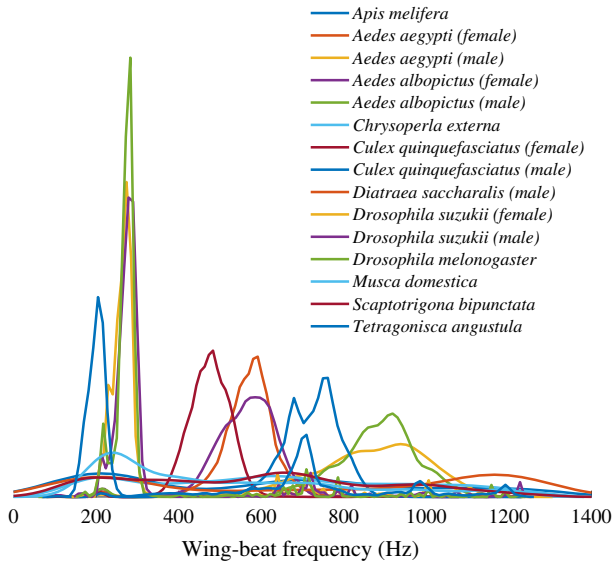
Fig. 17 Density functions that fit the histograms of different insect species

in this variable. Each temperature was measured in Celsius degrees and rounded to the nearest integer value. Thereby, we ordered the observations of the examples over time in the stream following different patterns of change in temperature while hiding this variable from the dataset. We reiterate that although we have manipulated the sequence of the examples to control the drifts, all these changes are feasible in the real use of the sensor on dynamic environments. Additionally, for each temperature, we uniformly sampled examples that were collected within that temperature. As a result, we eliminate all other sources of drift beside the changes in temperature. Finally, in addition to sampling from individual temperatures, we also vary the proportion of the classes over time, to mimic natural influences in the activity of insects: circadian rhythm, the presence of predators, among others. We consider the following changes, which also name our datasets:

– **Incremental** In this pattern, the instances are arranged so that the temperature values are incrementally increased from 20 °C to 40 °C over all the stream;
– **Abrupt.** We consider five sudden change points in this pattern. The first instances of the stream were collected at a temperature of 30 °C, and then they abruptly change to 20 °C. After a time, the temperature back to change for values around 35 °C. Similarly, other three abrupt changes occur until the end of the stream;
– **Incremental-gradual** In this pattern, the observed temperature in the first instances is around 37 °C and incrementally decrease until 35 °C. For a period, we have a gradual change where the temperature of the instances intercalates in the values of 35 °C and 23 °C until definitively change for 23 °C. In this period, two different concepts are active at the same time. At the end of the stream, the temperature back to incrementally increase until 27 °C;
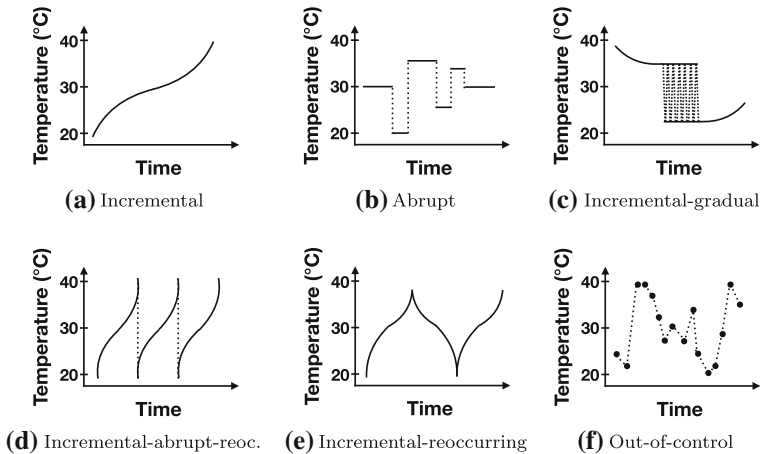
Fig. 18 Patterns of changes given the variable temperature to build the insect stream data

– **Incremental-abrupt-reoccurring** This pattern provides three recurrent cycles of incremental changes where the temperature increase from 20 to 40 °C. Between the end and beginning of a cycle of incremental changes, we have an abrupt change;

– **Incremental-reoccurring** In this pattern, there exist three cycles of incremental changes over time. In the first cycle, the temperature increases from 20 to 40 °C. In the second cycle, the temperature decreases from 40 to 20 °C. In the end, the temperature turns to increase to 40 °C. Although the stream presents two clear recurrent patterns where the values are increased, we also can consider the cycle of decreasing temperature as recurrent, but in an "inverse" arrival order of the instances;

– **Out-of-control** In this case, we have a lack of pattern in the occurrence of changes in the temperature. It means that is expected the arrival over time of instances observed at any temperature. This dataset is composed of all collected data in uniformly random order. As each example is sampled uniformly sampled at each time during the stream, this dataset must be drift-free.

Figure 18 graphically illustrates the patterns of changes presented in our datasets.

For the first five patterns showed in Fig. 18, we built two datasets for each one, being the first with balanced and the second with imbalanced class distribution. For the last dataset (Out-of-control), we have only an imbalanced version. Thus, we have a total of 11 different datasets.

Figure 19 shows the distribution over 24 class labels from the Out-of-control dataset. In this dataset, the *tet-angustula* and *musca* are the majority classes with 170,220 (18.81%) and 168,819 (18.65%) instances, respectively. While classes such as *psilid* and *cx-tarsalis-male* are the minority classes with only 17 and 157 instances, respectively. Thus, this dataset has two main challenges: the lack of a pattern to distinguish the concepts and overcome the temporal overlap, and imbalanced distribution. One additional note is that the proportions of the classes in the dataset are subject to data collection bias and do not represent the real proportion of the species in nature. Fur-
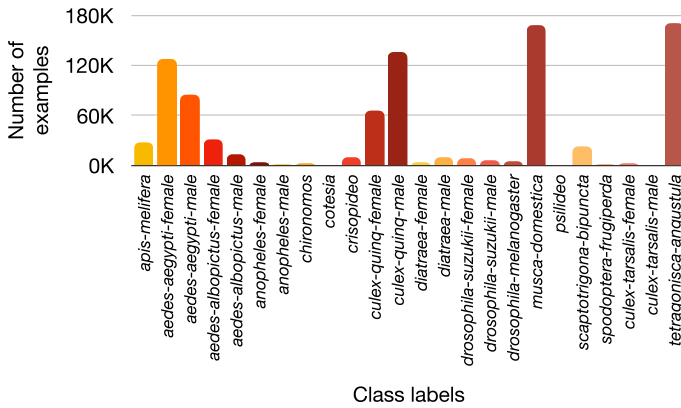
Fig. 19 Class distribution of out-of-control dataset

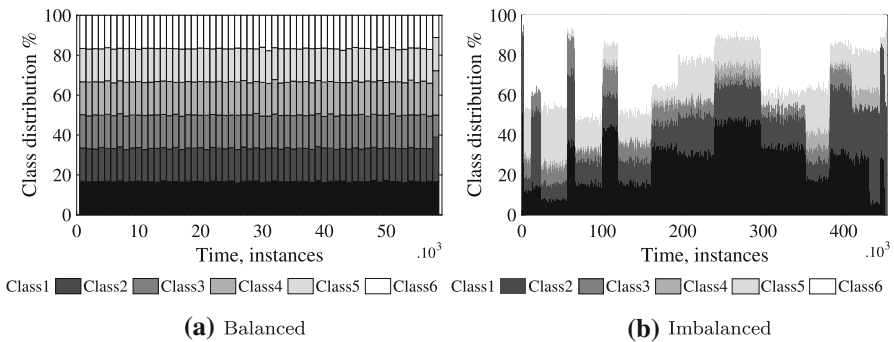**(a)** Balanced   **(b)** Imbalanced

Fig. 20 Changes in class proportion for the Incremental dataset considering the balanced and imbalanced class data versions. Each bar in the plots represents the class proportions into a window with 1000 consecutive examples in the stream

thermore, we expect such proportions to vary according to time, region, and ambient conditions.

Figure 20 illustrates the changes in the classes proportion over time for the two versions of the Incremental dataset. However, we note that not all balanced data versions are as well behaved as seen in Fig. 20a.

All datasets have 33 features, as previously discussed in Sect. 5.3. Except for the Out-of-control dataset that has 24 class labels, all other datasets have 6 class labels related to the species *Aedes aegypti* (female and male), *Aedes albopictus* (female and male), and *Culex quinquefasciatus* (female and male). The 24 class labels from the Out-of-control dataset can be seen in Fig. 19. Besides the higher number of class labels of this data, another interesting characteristic is the emergence of new classes over time, which allows their use in the evaluation of approaches for novelty detection (Masud et al. 2009). As is also often the disappearance of certain classes over time, this dataset can be useful for assessing solutions dealing with significant changes in $P(Y)$.

Table 2 Description of the insect stream datasets

Dataset	Instances	Change point(s)
Incremental (bal.)	57,018	Throughout all the stream
Incremental (imbal.)	452,044	Throughout all the stream
Abrupt (bal.)	52,848	14,352; 19,500; 33,240; 38,682; 39,510
Abrupt (imbal.)	355,275	83,859; 128,651; 182,320; 242,883; 268,380
Incremental-gradual (bal.)	24,150	14,028
Incremental-gradual (imbal.)	143,323	58,159
Incremental-abrupt-reoccurring (bal.)	79,986	26,568; 53,364
Incremental-abrupt-reoccurring (imbal.)	452,044	150,683; 301,365
Incremental-reoccurring (bal.)	79,986	26,568; 53,364
Incremental-reoccurring (imbal.)	452,044	150,683; 301,365
Out-of-control	905,145	Throughout all the stream

In Table 2, we show a description of the datasets as the number of instances and the position of the change points. Their names can identify the patterns of changes for each dataset.

5.6 Temporal overlap

Interesting datasets for streaming problems include different aspects discussed in this article: changes in the proportions of the classes over time, changes in the distribution of the features within each class over time, and temporal overlap, i.e., the dynamism of the overlap depending on which concept is responsible for the current examples.

We showed in the previous sections that the data we are providing have changes in the distribution of the features as we vary the temperature, and also a great deal of overlap between classes for at least the wing-beat frequency attribute. We consider our concept to relate to the temperature directly, and, for all but one version of the dataset, the temperature is the hidden variable that evolves while being stable within windows in the stream. For that reason, temperature overlap coincides with temporal overlap, and the former is the source of the latter.

A relevant question is whether there is a smaller overlap between the classes when we consider data for each temperature value than the data with all temperatures together. If that is not the case, we may need not worry about forgetting mechanisms to discard old data, since new concepts are likely to occupy empty regions in the feature space, as the temperature varies and we aggregate more data over time. However, if class overlap varies, a classification system can potentially benefit from identifying boundaries between different concepts and using models specifically trained for each one of them.

To illustrate this idea, consider a subset of the data that contains only female *Aedes aegypti* and female *Culex quinquefasciatus*. Each temperature was measured in Celsius degrees and rounded to the nearest integer. We sampled 2500 examples from each
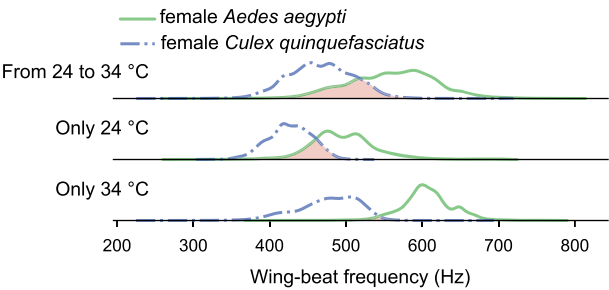
Fig. 21 Illustration of a case of temporal overlap. When we can discriminate the data according to the current temperature, we have smaller class overlap in the wing-beat frequency

Table 3 Values for a case of temporal overlap. When we can discriminate the data according to the current temperature, we have smaller class overlap in the wing-beat frequency

Temperature (°C)	24	26	28	30	32	34
Overlap (%)	29	32	28	23	19	5

Table 4 Indirect effect of temporal overlap. When we can discriminate the data according to the current temperature, we have higher accuracy for the insect data

Temperature (°C)	24	26	28	30	32	34
Individual classifiers	86	87	88	89	92	98
Single classifier	66	81	88	87	89	93

species for each one of the following temperatures (in Celsius): 24, 26, 28, 30, 32, and 34.

In Fig. 21, we visually illustrate the difference in class overlap for wing-beat frequency. This illustration is complemented by Table 3, which presents the numerical overlap between the two classes for each temperature. The overlap when all temperatures are considered together is 36%, while the average overlap when each temperature is isolated is 23%. The overlaps were estimated by taking the minimum between histograms with 100 bins.

Finally, to not limit ourselves to only one feature (wing-beat frequency), we indirectly measured the effect of the difference of overlaps by evaluating a classification task. We compared the use of individual classifiers for each temperature against a single classifier trained with data from all temperatures. The accuracy rates were obtained via 10-fold cross-validation and Random Forest classifier with 200 trees. We used all 33 features from the insect data. Table 4 presents the accuracy rates obtained. The single classifier achieves 84% accuracy for the whole data, while individual classifiers average 90%. We note that greater differences can be observed depending on which temperature is individually assessed: some temperatures apparently suffer a greater deal with temporal overlap than other ones. One example is 24 °C. It is the most difficult case even with an individual classifier, and is also the most harmed by the use of a conjoint classifier, with a 20% difference in their accuracy.

6 USP data stream repository

Aiming to mitigate possible flaws in the experimental evaluation of future proposals on stream learning due to the lack of real-world data, we provide to the machine learning community a new public repository called USP Data Stream Repository.[2] In this repository, we make available 27 datasets from different real problems composed by 16 data previously evaluated by other works from literature and 11 new datasets obtained by the optical sensor for automatic insect recognition.[3] It is important to note that we want to feed this repository regularly with new data from collaborative contributions.

We suggest that stream classifiers and drift detector algorithms should be tested on a wide range of datasets, mainly the real ones to avoid biased conclusions. It is a usual practice in more consolidated areas, such as machine learning in general (Dua and Graff 2017) and time-series (Chen et al. 2015), which contributes to the research advancement and maturity of the data stream area. At the same time, comparisons against baseline methods such as those proposed by Bifet et al. (2013) are also essential for the better performance analysis of new proposals. In this direction, we include in the repository the results achieved by two simple baselines methods for all datasets.

7 Evaluation and discussion

Besides to provide benchmark datasets to evaluate classifiers and drift detection methods, we also report the results achieved by state-of-the-art methods in our proposed data. The availability of benchmark data accompanied by the results achieved by methods from literature aims to make experiments from different researchers from the data stream community easily comparable and reproducible.

We run all experiments of stream classification and drift detection using the MOA framework software (Bifet et al. 2010a), which contains implementations of several state-of-the-art methods.

7.1 Classification

In the experimental evaluation of the classification task, we consider two naive baseline classifiers (Bifet et al. 2013): (i) No-Change and (ii) Majority-Class. Both approaches do not use any input attributes and classify only using past label information. The No-Change classifier ever predicts the next class label as the same of last seen class label. The Majority-Class made their prediction based on the majority class of a moving window over the stream with 1000 instances.

In addition to the baseline classifiers, we also evaluate the following stream algorithms: (i) incremental Naive Bayes (NB), (ii) Very Fast Decision Trees (Hulten et al. 2001) with Naive Bayes classifiers at the leaves, (iii) Leveraging Bagging with 10 VFDT in the ensemble (Bifet et al. 2010b), and (iv) Adaptive Random Forest (Gomes

---

[2] Available online at https://sites.google.com/view/uspdsrepository.

[3] The datasets are encrypted under the following password: DMKD2018.

Table 5 Prequential accuracy achieved by state-of-the-art methods in the Insect Stream Data

Dataset	No-change	Maj.Class	NB	VFDT	Lev.Bag.	ARF
Inc (bal.)	16.04	11.51	47.37	45.65	61.42	64.29
Inc (imbal.)	28.23	29.76	49.30	44.92	75.13	78.94
Abrupt (bal.)	28.98	16.07	50.77	49.85	68.39	74.34
Abrupt (imbal.)	29.15	28.49	52.18	48.46	72.28	80.02
Inc-gradual (bal.)	38.43	15.76	52.32	51.85	72.51	77.92
Inc-gradual (imbal.)	30.16	29.52	57.46	53.36	73.21	79.35
Inc-abrupt-reoc (bal.)	42.39	16.65	58.55	58.39	70.91	74.95
Inc-abrupt-reoc (imbal.)	28.16	29.76	52.34	51.03	69.13	77.60
Inc-reoc (bal.)	40.46	16.66	48.77	47.83	72.30	77.13
Inc-reoc (imbal.)	28.21	29.76	52.58	55.22	69.56	77.62
Out-of-control	13.06	18.80	45.99	44.70	53.58	70.45

et al. 2017). We based our choices on the efficiency and popularity of the methods available for evaluation.

We consider prequential evaluation (Gama et al. 2013) over a sliding window of 1,000 instances to evaluate the classification performance of the algorithms. In Table 5, we show the results achieved by the methods.

For all datasets, we can note that the Adaptive Random Forest (ARF) presented the best overall results, followed by the Leveraging Bagging (Lev.Bag.). For these methods, the overall results are around 70-80% for different patterns of drifts, which are slightly inferior when compared with our previous evaluations on static data with a similar feature set in the problem of insect species recognition (Souza et al. 2013; Silva et al. 2015; Qi et al. 2015). As expected, both incremental algorithms (VFDT and Naive Bayes), which do not consider a strategy to deal with concept drifts explicitly, were outperformed by more powerful data stream classifiers. The poor performance of baseline classifiers gives us empirical evidence that undesirable characteristics such as temporal dependence and the prevalence of majority classes are underrepresented in our data.

Table 5 provides a general view of the classification performance of the algorithms. However, in data streams, we are frequently interested in seeing these performances over time. Besides, the performance over time of some approaches, such as the baseline classifiers can help to understand the changes in the data. In this direction, we present below the individual evaluation for each one of datasets from our benchmark.

Figure 22 shows the prequential accuracy results achieved over time by the compared methods for the balanced and imbalanced versions of Incremental data. Given the slow speed of the incremental changes, the algorithms tend to present more stable performances without significant accuracy increase or decrease. In the imbalanced version, the algorithms show instabilities in well-defined points, probably due to the $P(Y)$ changes.

Figure 23 shows the results over time for the two versions of Abrupt data. In the balanced version, we can note the presence of temporal dependence in four different
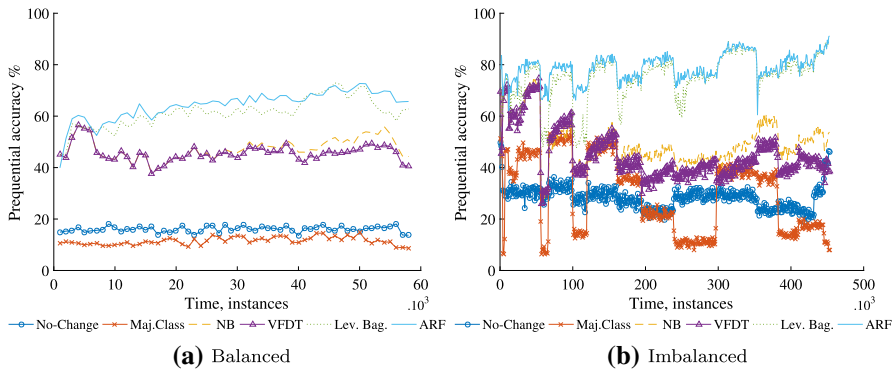
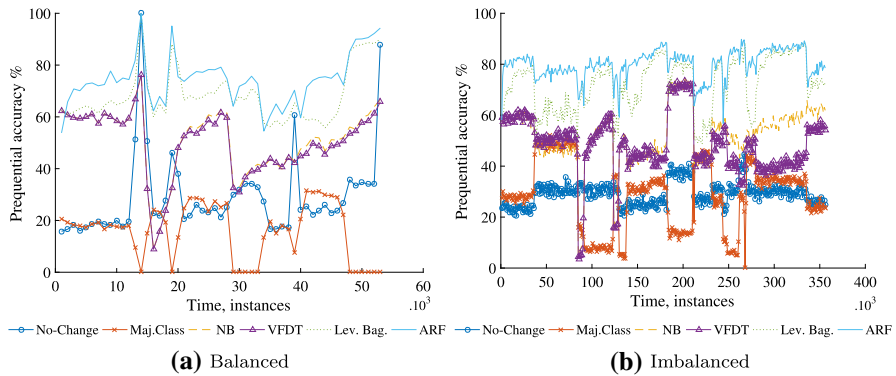**Fig. 22** Prequential accuracy on the incremental data

Fig. 23 Prequential accuracy on the abrupt data

points of the stream (close to the times 14,000; 19,000; 40,000; and 52,000). However, in all cases, they are rapidly dissolved as we can note by the poor performance of No-Change classifier over time.

Figure 24 shows the results over time for the Incremental-gradual data. For the balanced version, we see a drastic fall in the performances of the classifiers immediately before 15,000 instances from the stream, which is related to the occurrence of gradual drift. At this period, the stream presents instances from two different concepts at the same time until the change for a new concept is complete.

Figure 25 shows the results for the Incremental-abrupt-reoccurring data. In the balanced version, we can note three different periods where the classifiers achieve accuracy peaks in their performances, followed by a significant fall in the first two cases. These periods correspond to the end of a cycle of incremental changes and the start of an abrupt change. In the imbalanced data version, the analysis of VFDT results can help to understand the data better. In this case, we can note in three different periods, an incremental fall of the classifier performance with values between 80 and 40%.
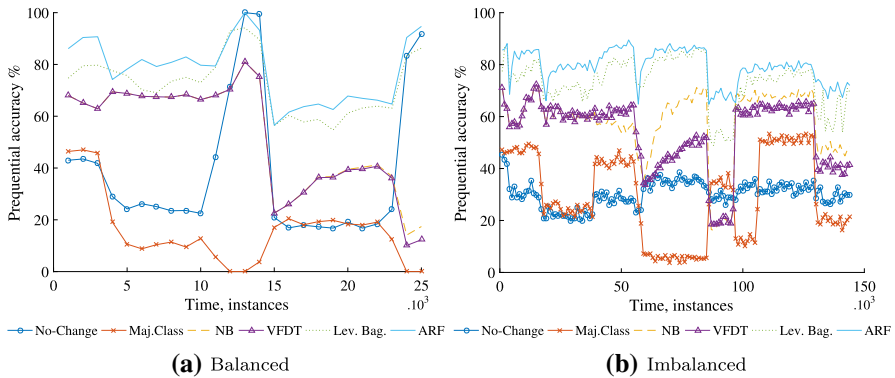
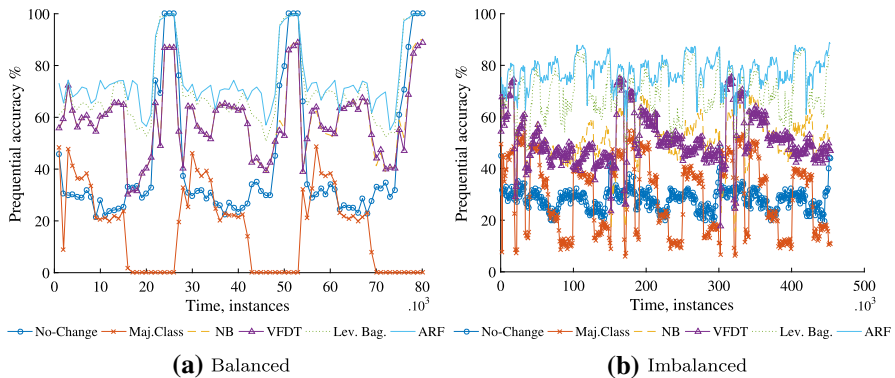**Fig. 24** Prequential accuracy on the incremental-gradual data

Fig. 25 Prequential accuracy on the incremental-abrupt-reoccurring data

In Fig. 26, we show the results for balanced and imbalanced versions of Incremental-reoccurring data. Although the main difference of this dataset with Incremental-abrupt-reoccurring data is the presence of abrupt changes at two different times, the results are very similar to those previously shown in Fig. 25. It can mean that abrupt changes are not responsible for significant impacts in the performances of the algorithms, mainly when we observe recurring concepts in the stream. In general, the algorithms have more difficult to adapt for incremental changes.

Figure 27 shows the results for the Out-of-control data. It is interesting to note that although this dataset has a large number of class labels and undefined changes in type and number, the classifiers show more stable performances over time when compared with other datasets. However, the results are limited. For example, the best classifier (ARF), shows an overall prequential accuracy around to 70%.
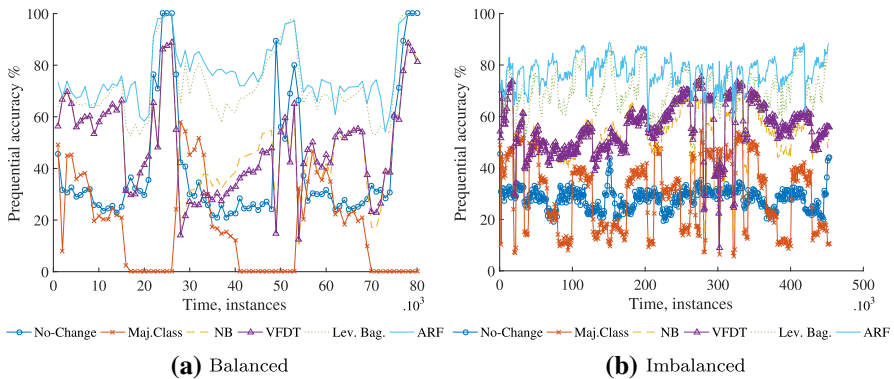
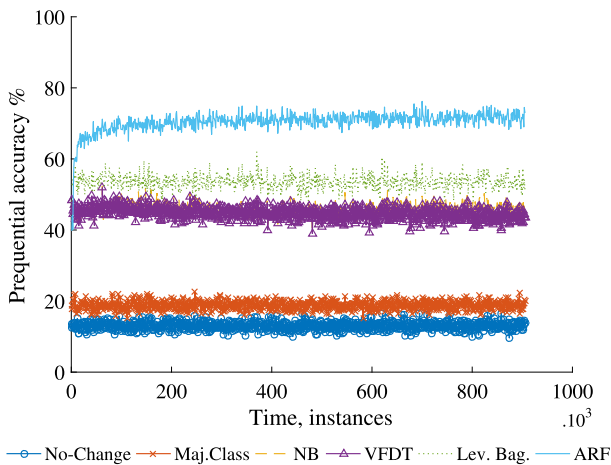**Fig. 26** Prequential accuracy on the incremental-reoccurring data

Fig. 27 Prequential accuracy on the out-of-control data

7.2 Drift detection

We choose representative methods for different drift detection approaches to evaluate the performance of detectors considering our benchmark data. Specifically, we consider the following methods:

– Sequential analysis: Page–Hinkley Test (PHT) and CUSUM (Page 1954);
– Statistical process control: Drift Detection Method (DDM) (Gama et al. 2004) and Exponentially Weighted Moving Average (EWMA) (Ross et al. 2012);
– Comparison of data distributions: Adaptive Windowing (ADWIN) (Bifet and Gavalda 2007), SEED (Huang et al. 2014), and Statistical Test of Equal Proportions (STEPD) (Nishida and Yamauchi 2007).

Regarding the parameters of the detectors, we consider a window size with 1,000 examples and a minimum of 100 examples to detect a drift. For a fair comparison, the remaining parameters follow the default values suggested by MOA. As all evaluated

methods require a base classifier, we consider the Naive Bayes for all approaches to standardize the experimental evaluation. Thus, we can evaluate the performance of drift detection methods based on the prequential accuracy without the influence of the classification algorithm.

In Table 6, we show the overall prequential accuracy (Acc.) and the total of changes detected (C.D.) by the different drift detectors evaluated considering the Insect Stream Data. For each dataset, we highlighted the best accuracy in **bold**. In general, the best results are achieved by the methods ADWIN and STEPD.

We chose three different cases to better analyze the drift detection task over time. Specifically, we present the results of STEPD, a method based on the comparison of distributions composed by data of two accuracies achieved by the classifier in two times: the recent one and the overall one. The balanced data versions of Abrupt, Incremental-abrupt-reoccurring, and Incremental-reoccurring datasets were analyzed.

In Fig. 28, we show the prequential accuracy achieved by the Naive Bayes classifier using the STEPD drift detector method on Abrupt data. In this figure, we can see the 28 change points detected in the vertical red lines. In this data, we have six different concepts that occur after five abrupt changes. Different background colors in the figure represent the six concepts (A–F) of this data.

We can note in Fig. 28 that even during the arrival of instances from a stable concept, the method incorrectly detects different change points. In most cases, the model adaptations in these points do not lead to better accuracy, except in the last changes identified into the concepts B, D, and F. We also can note that all abrupt changes were correctly identified.

Similarly, in Fig. 29 we present the results on Incremental-abrupt-reoccurring data. In this data, we have two different points with well defined abrupt changes. However, it also occurs minor incremental changes between the abrupt changes. The gradient in the background color of the figure represents the incremental changes. All the changes are indicated in the top view of the figure.

Given the constant occurrence of incremental changes in these data over all the stream, we can note a high number of change points identified by the method in Fig. 29. Specifically, STEPD identified 61 change points.

In Fig. 30, we show the results on Incremental-reoccurring data. As this data only present incremental changes over time, it is more difficult to precisely indicate the change points in the stream. However, we show a general view of these changes by the gradient in the background color of the figure. In this dataset, STEPD identified 59 change points.

8 Conclusions

In this paper, we discuss the challenges faced by the stream learning community concerning the reduced number of real-world data and the lack of a benchmark to evaluate adaptive classifiers and drift detectors. This gap leads to the use of synthetic data accompanied by a small number of real data in the evaluation of new proposals. The main problem of this practice is the possibility of data bias, which can lead to incorrect conclusions about stream algorithms behavior. We also present a review

Table 6 Overall prequential accuracy (Acc.) and total of changes detected (CD) by different drift detectors

| Dataset	ADWIN		PHT		CUSUM		DDM		EWMA		SEED		STEPD	
	Acc.	CD	Acc.	CD	Acc.	CD	Acc.	CD	Acc.	CD	Acc.	CD	Acc.	CD
Inc (bal.)	52.68	3	54.17	1	**56.63**	5	52.72	1	47.37	0	54.96	9	56.55	30
Inc (imbal.)	**61.02**	136	58.79	42	59.97	99	49.32	9	50.37	104	58.50	86	59.96	225
Abrupt (bal.)	62.48	7	62.14	6	64.63	8	60.36	5	65.40	90	65.73	21	**66.02**	28
Abrupt (imbal.)	58.72	94	59.89	37	60.70	71	56.78	9	52.23	85	60.31	68	**61.52**	185
Inc-gradual (bal.)	**72.26**	6	68.30	7	69.20	9	65.40	6	71.39	39	70.38	14	71.51	25
Inc-gradual (imbal.)	**67.70**	36	62.57	20	63.53	41	55.25	15	58.90	182	62.30	64	62.32	64
Inc-abrt-reoc (bal.)	63.80	22	63.25	17	65.12	25	61.35	16	66.23	114	67.90	60	**68.77**	61
Inc-abrt-reoc (imbal.)	59.98	157	58.51	90	59.15	120	53.13	31	51.41	76	58.79	199	**60.22**	297
Inc-reoc (bal.)	65.93	26	64.59	16	65.87	21	63.96	21	66.45	108	**69.82**	47	69.45	59
Inc-reoc (imbal.)	**60.39**	152	58.16	67	59.65	122	55.13	34	51.91	96	59.00	163	59.68	242
Out-of-control	**49.92**	237	47.22	52	48.40	155	45.75	3	45.99	0	46.86	98	48.83	444
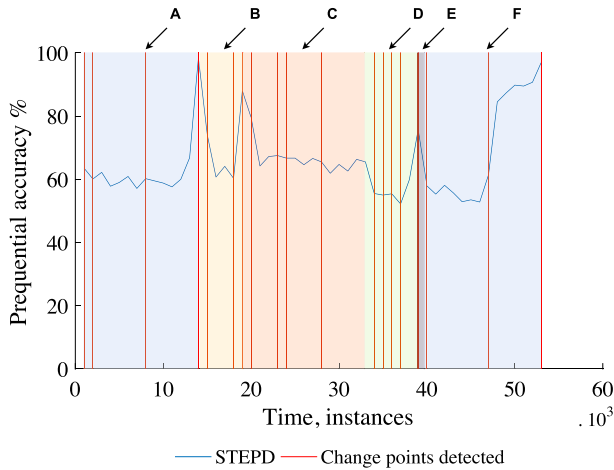
Fig. 28 Prequential accuracy of Naive Bayes classifier with STEPD drift detection method and the change points detected considering the Abrupt data (balanced). Different background colors represent the six concepts (A–F) of this data (Color figure online)
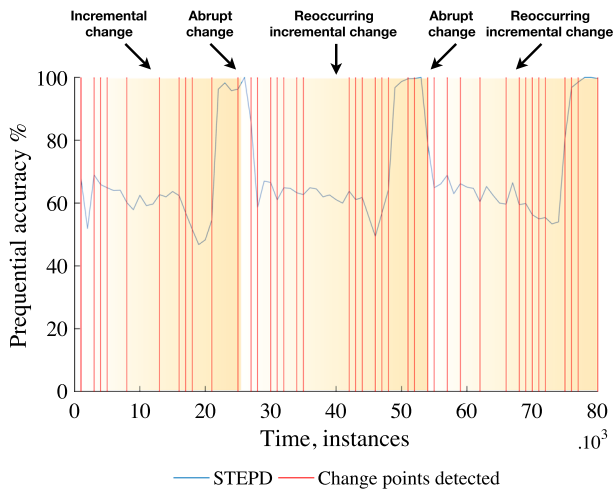

Fig. 29 Prequential accuracy of Naive Bayes classifier with STEPD drift detection method and the change points detected considering the incremental-abrupt-reoccurring data (balanced). The gradient in the background color represents the incremental changes. The abrupt changes occur between two consecutive incremental changes. All the changes are indicated in the top view of the figure (Color figure online)

regarding the main real datasets evaluated in the literature and discuss some weaknesses in such data as the lack of knowledge about the type/pattern of change and when it occurs in the stream.

To mitigate some of the problems identified in the evaluation of stream methods concerning the lack of real data, we propose the use of 11 new datasets collected by an optical sensor that measures the flying behavior of insects. This data is used in a relevant application of public health related to the use of a Smart Trap to attract and
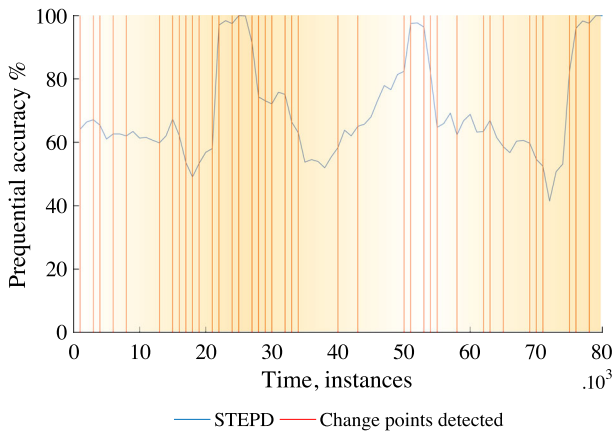
Fig. 30 Prequential accuracy of Naive Bayes classifier with STEPD drift detection method and the change points detected considering the incremental-reoccurring data (balanced). The gradient in the background color represents the incremental changes (Color figure online)

capture target species such as the vector of diseases. In this application, non-stationary data are generated over time in streaming fashion due to changes in the environment, which impacts the insects' behavior. Our proposed data has interesting characteristics to be explored by researchers, such as different patterns of changes (incremental, abrupt, gradual, and reoccurring), indicators of the presence of each change and when they occur, the presence of complex changes in the class distribution, a significant number of instances, among others.

Although the proposed benchmark constitutes an essential contribution to the stream mining community, it is also important to note that such data also have some limitations. We highlighted two of them. First, to precisely indicate the drift points and the types of drift, we performed a manual manipulation in the original arrival order of the examples. Also, to avoid problems such as temporal dependence, we performed a shuffling procedure into a window with similar examples. In practice, such procedures do not affect the meaning of the application, which can experience the simulated changes in real environments. However, such manipulation could be interpreted as responsible for generating data *semi-real* or not entirely real. The second limitation, which most of the datasets from literature also presents, is the lack of time-stamps. Such limitation poses some restrictions to evaluate issues where the time is an additional constraint factor in the learning task. For example, with the time-stamps, it is possible to verify if the classification model is updated at the available time between the examples' arrival. Also, the algorithms can take this time into consideration to perform other updates in idle periods of the classifier.

We also provide to the machine learning community a new public repository called USP Data Stream Repository, where we make available 27 datasets from different real problems composed by 16 data previously evaluated by other works from literature and 11 new datasets obtained by the optical sensor for automatic insect recognition. In this repository, we also present the results achieved by two baseline methods for

all datasets. This repository will be regularly fed with new data from our future works and donation.

References

Ajtai M (1988) The complexity of the pigeonhole principle. In: Annual symposium on foundations of computer science, pp 346–355

Alippi C, Roveri M (2008) Just-in-time adaptive classifiers—part I: detecting nonstationary changes. IEEE Trans Neural Netw 19(7):1145–1153

Alon N, Matias Y, Szegedy M (1999) The space complexity of approximating the frequency moments. J Comput Syst Sci 58(1):137–147

Baena-Garcia M, del Campo-Avila J, Fidalgo R, Bifet A, Gavalda R, Morales-Bueno R (2006) Early drift detection method. In: International workshop on knowledge discovery from data streams, pp 77–86

Bagnall A, Lines J, Vickers W, Keogh E (2019) The UEA & UCR time series classification repository. www.timeseriesclassification.com. Accessed Mar 2020

Bartholomay LC, Waterhouse RM, Mayhew GF, Campbell CL, Michel K, Zou Z, Ramirez JL, Das S, Alvarez K, Arensburger P et al (2010) Pathogenomics of *Culex quinquefasciatus* and meta-analysis of infection responses to diverse pathogens. Science 330(6000):88–90

Batista GEAPA, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor Newsl 6(1):20–29

Batista G, Keogh EJ, Mafra-Neto A, Rowton E (2011) SIGKDD demo: sensors and software to allow computational entomology, an emergingapplication of data mining. In: ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), pp 761–764

Ben-David S, Blitzer J, Crammer K, Pereira F (2007) Analysis of representations for domain adaptation. In: Advances in neural information processing systems, pp 137–144

Bifet A (2009) Adaptive learning and mining for data streams and frequent patterns. SIGKDD Explor Newsl 11(1):55–56

Bifet A, Holmes G, Kirkby R, Pfahringer B (2010a) MOA: massive online analysis. J Mach Learn Res 11(May):1601–1604

Bifet A, Gavalda R (2007) Learning from time-changing data with adaptive windowing. In: SIAM international conference on data mining (SDM), pp 443–448

Bifet A, Holmes G, Pfahringer B (2010b) Leveraging bagging for evolving data streams. In: Joint European conference on machine learning and knowledge discovery in databases (ECML), pp 135–150

Bifet A, Holmes G, Pfahringer B, Kirkby R, Gavaldà R (2009) New ensemble methods for evolving data streams. In: ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 139–148

Bifet A, Read J, Zliobaite I, Pfahringer B, Holmes G (2013) Pitfalls in benchmarking data stream classification and how to avoid them. In: Joint European conference on machine learning and knowledge discovery in databases, pp 465–479

Bifet A, Zhang J, Fan W, He C, Zhang J, Qian J, Holmes G, Pfahringer B (2017) Extremely fast decision tree mining for evolving data streams. In: ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 1733–1742

Blackard JA, Dean DJ (1999) Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. Comput Electron Agric 24(3):131–151

Breiman L, Friedman J, Olshen RA, Stone CJ (1984) Classification and regression trees. Chapman and Hall/CRC Press, London

Brzezinski D, Stefanowski J (2011) Accuracy updated ensemble for data streams with concept drift. In: International conference on hybrid artificial intelligence systems, pp 155–163

Cattral R, Oppacher F, Deugo D (2002) Evolutionary data mining with automatic rule generalization. Recent Adv Comput Comput Commun 1(1):296–300

Cha SH, Srihari SN (2002) On measuring the distance between histograms. Pattern Recogn 35(6):1355–1370

Chadwick LE, Williams CM (1949) The effects of atmospheric pressure and composition on the flight of drosophila. Biol Bull 97(2):115–137

Chaudhuri S, Motwani R, Narasayya V (1999) On random sampling over joins. ACM SIGMOD Record 28(2):263–274

Chawla NV, Japkowicz N, Kotcz A (2004) Special issue on learning from imbalanced data sets. ACM SIGKDD Explor Newsl 6(1):1–6

Chen S, He H (2011) Towards incremental learning of nonstationary imbalanced data stream: a multiple selectively recursive approach. Evol Syst 2(1):35–50

Chen Y, Why A, Batista GEAPA, Mafra-Neto A, Keogh E (2014) Flying insect classification with inexpensive sensors. J Insect Behav 27(5):657–677

Chen Y, Keogh E, Hu B, Begum N, Bagnall A, Mueen A, Batista GEAPA (2015) The UCR time series classification archive. www.cs.ucr.edu/~eamonn/time_series_data/. Accessed Oct 2018

Cutwa MM, O'Meara GF (2006) Photographic guide to common mosquitoes of Florida. University of Florida, Florida Medical Entomology Laboratory

Dasu T, Krishnan S, Venkatasubramanian S, Yi K (2006) An information-theoretic approach to detecting changes in multi-dimensional data streams. In: Symposium on the interface of statistics, computing science, and applications

Datar M, Gionis A, Indyk P, Motwani R (2002) Maintaining stream statistics over sliding windows. In: ACM-SIAM symposium on discrete algorithms, pp 635–644

Demsar J, Bosnic Z (2018) Detecting concept drift in data streams using model explanation. Expert Syst Appl 92:546–559

Ditzler G, Polikar R (2013) Incremental learning of concept drift from streaming imbalanced data. Trans Knowl Data Eng 25(10):2283–2301

Ditzler G, Roveri M, Alippi C, Polikar R (2015) Learning in nonstationary environments: a survey. IEEE Comput Intell Mag 10(4):12–25

Domingos P (2012) A few useful things to know about machine learning. Commun ACM 55(10):78–87

Domingos P, Hulten G (2000) Mining high-speed data streams. In: ACM SIGKDD international conference on Knowledge discovery and data mining, pp 71–80

Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed Mar 2020

Dyer KB, Capo R, Polikar R (2014) Compose: a semisupervised learning framework for initially labeled nonstationary streaming data. IEEE Trans Neural Netw Learn Syst 25(1):12–26

Eisen L, Moore CG (2013) *Aedes* (*stegomyia*) *aegypti* in the Continental United States: a vector at the cool margin of its geographic range. J Med Entomol 50(3):467–478

Faithfull WJ, Rodríguez JJ, Kuncheva LI (2019) Combining univariate approaches for ensemble change detection in multivariate data. Inf Fusion 45:202–214

Fawcett T, Flach PA (2005) A response to Webb and Ting's on the application of ROC analysis to predict classification performance under varying class distributions. Mach Learn 58(1):33–38

Gama J (2010) Knowledge discovery from data streams. Chapman and Hall/CRC Press, London

Gama J, Gaber M (2007) Learning from data streams: processing techniques in sensor networks. Springer, Berlin

Gama J, Sebastião R, Rodrigues PP (2013) On evaluating stream learning algorithms. Mach Learn 90(3):317–346

Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Comput Surv 46(4):44

Gama J, Medas P, Castillo G, Rodrigues P (2004) Learning with drift detection. In: Brazilian symposium on artificial intelligence, pp 286–295

Gama J, Medas P, Rodrigues P (2005) Learning decision trees from dynamic data streams. In: ACM symposium on Applied computing (SAC), pp 573–577

Ganti V, Gehrke J, Ramakrishnan R (1999) A framework for measuring changes in data characteristics. In: ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems (PDS), pp 126–137

Gebru A, Jansson S, Ignell R, Kirkeby C, Prangsma JC, Brydegaard M (2018) Multiband modulation spectroscopy for the determination of sex and species of mosquitoes in flight. J Biophoton 11(8):1–13

Ghazikhani A, Monsefi R, Yazdi HS (2013) Recursive least square perceptron model for non-stationary and imbalanced data stream classification. Evol Syst 4(2):119–131

Gilbert AC, Guha S, Indyk P, Kotidis Y, Muthukrishnan S, Strauss MJ (2002) Fast, small-space algorithms for approximate histogram maintenance. In: ACM symposium on theory of computing, pp 389–398

Goldenberg I, Webb GI (2019) Survey of distance measures for quantifying concept drift and shift in numeric data. Knowl Inf Syst 60:591–615

Gomes HM, Bifet A, Read J, Barddal JP, Enembreck F, Pfharinger B, Holmes G, Abdessalem T (2017) Adaptive random forests for evolving data stream classification. Mach Learn 106(9–10):1469–1495

Gonçalves PM Jr, de Carvalho Santos SG, Barros RS, Vieira DC (2014) A comparative study on concept drift detectors. Expert Syst Appl 41(18):8144–8156

González P, Castaño A, Chawla NV, Coz JJD (2017) A review on quantification learning. ACM Comput Surv 50(5):74

Gratz N (2004) Critical review of the vector status of *Aedes albopictus*. Med Vet Entomol 18(3):215–227

Harries M (1999) Splice-2 comparative evaluation: Electricity pricing. Technical Report 1, University of New South Wales, Sydney, Australia

Harries MB, Sammut C, Horn K (1998) Extracting hidden context. Mach Learn 32(2):101–126

Hoens TR, Polikar R, Chawla NV (2012) Learning from streaming data with concept drift and imbalance: an overview. Prog Artif Intell 1(1):89–101

Hotelling H (1992) The generalization of student's ratio. In: Kotz S, Johnson NL (eds) Breakthroughs in statistics: foundations and basic theory. Springer, New York, pp 54–65

Huang DTJ, Koh YS, Dobbie G, Pears R (2014) Detecting volatility shift in data streams. In: IEEE international conference on data mining (ICDM), pp 863–868

Hulten G, Spencer L, Domingos P (2001) Mining time-changing data streams. In: ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 97–106

Ikonomovska E, Gama J, Dveroski S (2011) Learning model trees from evolving data streams. Data Min Knowl Disc 23(1):128–168

Katakis I, Tsoumakas G, Banos E, Bassiliades N, Vlahavas I (2009) An adaptive personalized news dissemination system. J Intell Inf Syst 32(2):191–212

Kelly MG, Hand DJ, Adams NM (1999) The impact of changing populations on classifier performance. In: ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), pp 367–371

Keogh E, Kasetty S (2003) On the need for time series data mining benchmarks: a survey and empirical demonstration. Data Min Knowl Disc 7(4):349–371

Khamassi I, Sayed-Mouchaweh M, Hammami M, Ghedira K (2018) Discussion and review on evolving data streams and concept drift adapting. Evol Syst 9(1):1–23

Kifer D, Ben-David S, Gehrke J (2004) Detecting change in data streams. In: International conference on very large data bases (VLDB), pp 180–191

Killourhy K, Maxion R (2010) Why did my detector do that?! In: International workshop on recent advances in intrusion detection, pp 256–276

Klinkenberg R (2004) Learning drifting concepts: example selection vs. example weighting. Intell Data Anal 8(3):281–300

Klinkenberg R, Joachims T (2000) Detecting concept drift with support vector machines. In: International conference on machine learning (ICML), pp 487–494

Krawczyk B, Minku LL, Gama J, Stefanowski J, Woźniak M (2017) Ensemble learning for data stream analysis: a survey. Inf Fusion 37:132–156

Kull M, Flach P (2014) Patterns of dataset shift. In: First international workshop on learning over multiple contexts at ECML-PKDD, pp 1–10

Kuncheva LI, Sánchez JS (2008) Nearest neighbour classifiers for streaming data with delayed labelling. In: IEEE international conference on data mining (ICDM), pp 869–874

Kuncheva LI (2013) Change detection in streaming multivariate data using likelihood detectors. IEEE Trans Knowl Data Eng 25(5):1175–1180

Linhart C, Harari G, Abramovich S, Buchris A (2009) PAKDD data mining competition 2009: New ways of using known methods. In: Pacific-Asia conference on knowledge discovery and data mining, pp 99–105

Li R, Wang S, Deng H, Wang R, Chang KCC (2012) Towards social user profiling: unified and discriminative influence model for inferring home locations. In: ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 1023–1031

Losing V, Hammer B, Wersing H (2015) Interactive online learning for obstacle classification on a mobile robot. In: International joint conference on neural networks, pp 1–8

Losing V, Hammer B, Wersing H (2016) KNN classifier with self adjusting memory for heterogeneous concept drift. In: IEEE International conference on data mining (ICDM), pp 291–300

Maletzke A, Reis DM, Cherman E, Batista GEAPA (2018) On the need of class ratio insensitive drift tests for data streams. In: International workshop on learning with imbalanced domains: theory and applications, pp 110–124

Maletzke A, Reis DM, Cherman E, Batista GEAPA (2019) DYS: a framework for mixture models in quantification. In: AAAI conference on artificial intelligence, pp 1–9

Manapragada C, Webb GI, Salehi M (2018) Extremely fast decision tree. In: ACM SIGKDD international conference on knowledge discovery & data mining (KDD), pp 1953–1962

Masud MM, Gao J, Khan L, Han J, Thuraisingham B (2009) Integrating novel class detection with classification for concept-drifting data streams. In: Joint European conference on machine learning and knowledge discovery in databases (ECML), pp 79–94

Matias Y, Vitter J, Wang M (2000) Dynamic maintenance of wavelet-based histograms. In: International conference on very large data bases, pp 101–110

Medlock JM, Hansford KM, Schaffner F, Versteirt V, Hendrickx G, Zeller H, Bortel WV (2012) A review of the invasive mosquitoes in europe: ecology, public health risks, and control options. Vector-Borne Zoonot Dis 12(6):435–447

Mellanby K (1936) Humidity and insect metabolism. Nature 138:124–125

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

Moreno-Torres JG, Raeder T, Alaiz-RodríGuez R, Chawla NV, Herrera F (2012) A unifying view on dataset shift in classification. Pattern Recogn 45(1):521–530

Mousson L, Dauga C, Garrigues T, Schaffner F, Vazeille M, Failloux AB (2005) Phylogeography of *Aedes* (*stegomyia*) *aegypti* (L.) and *Aedes* (*Stegomyia*) *albopictus* (*skuse*) (Diptera: Culicidae) based on mitochondrial DNA variations. Genet Res 86(1):1–11

Narasimhamurthy AM, Kuncheva LI (2007) A framework for generating data to simulate changing environments. In: International multi-conference: artificial intelligence and applications (IASTED), pp 384–389

Nishida K, Yamauchi K (2007) Detecting concept drift using statistical testing. In: International conference on discovery science, pp 264–269

Oh KJ, Kj K (2002) Analyzing stock market tick data using piecewise nonlinear model. Expert Syst Appl 22(3):249–255

Page ES (1954) Continuous inspection schemes. Biometrika 41(1/2):100–115

Pan SJ, Yang Q (2009) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359

Paupy C, Delatte H, Bagny L, Corbel V, Fontenille D (2009) *Aedes albopictus*, an arbovirus vector: from the darkness to the light. Microbes Infect 11(14–15):1177–1185

Qi Y, Cinar GT, Souza VMA, Batista GEAPA, Wang Y, Principe JC (2015) Effective insect recognition using a stacked autoencoder with maximum correntropy criterion. In: International joint conference on neural networks (IJCNN), pp 1–7

Quionero-Candela J, Sugiyama M, Schwaighofer A, Lawrence ND (2009) Dataset shift in machine learning. MIT Press, Cambridge

Ramamurthy S, Bhatnagar R (2007) Tracking recurrent concept drift in streaming data using ensemble classifiers. In: International conference on machine learning and applications (ICMLA), pp 404–409

Reis DM, Maletzke A, Batista GEAPA (2018a) Unsupervised context switch for classification tasks on data streams with recurrent concepts. In: ACM symposium on applied computing (SAC), pp 518–524

Reis DM, Flach P, Matwin S, Batista GEAPA (2016) Fast unsupervised online drift detection using incremental Kolmogorov–Smirnov test. In: ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 1545–1554

Reis DM, Maletzke A, Silva DF, Batista G (2018b) Classifying and counting with recurrent contexts. In: ACM SIGKDD international conference on knowledge discovery & data mining (KDD), pp 1983–1992

Rodriguez-Lujan I, Fonollosa J, Vergara A, Homer M, Huerta R (2014) On the calibration of sensor arrays for pattern recognition using the minimal number of experiments. Chemom Intell Lab Syst 130:123–134

Ross GJ, Adams NM, Tasoulis DK, Hand DJ (2012) Exponentially weighted moving average charts for detecting concept drift. Pattern Recogn Lett 33(2):191–198

Saenko K, Kulis B, Fritz M, Darrell T (2010) Adapting visual category models to new domains. In: European conference on computer vision. Springer, pp 213–226

Sarnelle J, Sanchez A, Capo R, Haas J, Polikar R (2015) Quantifying the limited and gradual concept drift assumption. In: International joint conference on neural networks, pp 1–8

Schlimmer JC, Granger RH (1986) Incremental learning from noisy data. Mach Learn 1(3):317–354

Shao J, Huang F, Yang Q, Luo G (2018) Robust prototype-based learning on data streams. IEEE Trans Knowl Data Eng 30(5):978–991

Shinkawa Y, Takeda SI, Tomioka K, Matsumoto A, Oda T, Chiba Y (1994) Variability in Circadian activity patterns within the *Culex pipiens* complex (Diptera: Culicidae). J Med Entomol 31(1):49–56

Silva DF, Souza VMA, Ellis DPW, Keogh EJ, Batista G (2015) Exploring low cost laser sensors to identify flying insect species. J Intell Robot Syst 80(1):313–330

Sobolewski P, Wozniak M (2013) Concept drift detection and model selection with simulated recurrence and ensembles of statistical detectors. J Univ Comput Sci 19(4):462–483

Souza VMA (2016) Classification of non-stationary data stream with application in sensors for insect identification. Ph.D. thesis, University of São Paulo

Souza VMA, Silva DF, Batista G (2013) Classification of data streams applied to insect recognition: Initial results. In: Brazilian conference on intelligent systems (BRACIS), pp 76–81

Souza VMA, Silva DF, Batista GEAPA, Gama J (2015a) Classification of evolving data streams with infinitely delayed labels. In: International conference on machine learning and applications (ICMLA), pp 214–219

Souza VMA, Silva DF, Gama J, Batista GEAPA (2015b) Data stream classification guided by clustering on nonstationary environments and extreme verification latency. In: SIAM international conference on data mining (SDM), pp 873–881

Souza VMA (2018) Asphalt pavement classification using smartphone accelerometer and complexity invariant distance. Eng Appl Artif Intell 74:198–211

Souza VMA, Giusti R, Batista AJL (2018) Asfault: a low-cost system to evaluate pavement conditions in real-time using smartphones and machine learning. Pervasive Mob Comput 51:121–137

Street WN, Kim YS (2001) A streaming ensemble algorithm (sea) for large-scale classification. In: ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 377–382

Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the KDD cup 99 data set. In: IEEE symposium on computational intelligence for security and defense applications (CISDA), pp 1–6

Taylor LR (1963) Analysis of the effect of temperature on insects in flight. J Anim Ecol 32(1):99–117

Tsymbal A (2004) The problem of concept drift: definitions and related work. Computer Science Department, Trinity College Dublin, Dublin, pp 1–7

Venkatasubramanian V, Rengaswamy R, Kavuri SN, Yin K (2003) A review of process fault detection and diagnosis: part III: process history based methods. Comput Chem Eng 27(3):327–346

Vergara A, Vembu S, Ayhan T, Ryan MA, Homer ML, Huerta R (2012) Chemical gas sensor drift compensation using classifier ensembles. Sens Actuators B: Chem 166:320–329

Villarreal SM, Winokur O, Harrington L (2017) The impact of temperature and body size on fundamental flight tone variation in the mosquito vector *Aedes aegypti* (Diptera: Culicidae): implications for acoustic lures. J Med Entomol 54(5):1116–1121

Vreeken J, Van Leeuwen M, Siebes A (2007) Characterising the difference. In: ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), pp 765–774

Wald A (1947) Sequential analysis. Wiley, Hoboken

Wang S, Minku LL, Yao X (2018) A systematic study of online class imbalance learning with concept drift. IEEE Trans Neural Netw Learn Syst 29(10):4802–4821

Wang S, Minku LL, Yao X (2013) A learning framework for online class imbalance learning. In: IEEE symposium on computational intelligence and ensemble learning, pp 36–45

Webb GI, Lee LK, Goethals B, Petitjean F (2018) Analyzing concept drift and shift from sample data. Data Min Knowl Disc 32(5):1179–1199

Widmer G, Kubat M (1996) Learning in the presence of concept drift and hidden contexts. Mach Learn 23(1):69–101

Yang Q, Wu X (2006) 10 challenging problems in data mining research. Int J Inf Technol Decis Mak 5(04):597–604

Zhu X (2010) Stream data mining repository.www.cse.fau.edu/~xqzhu/stream.html. Accessed Mar 2020

Zliobaite I (2010) Change with delayed labeling: when is it detectable? In: IEEE international conference on data mining workshops (ICDMW), pp 843–850

Zliobaite I (2013) How good is the electricity benchmark for evaluating concept drift adaptation. arXiv preprint arXiv:1301.3524

Zliobaite I (2011) Combining similarity in time and space for training set formation under concept drift. Intell Data Anal 15(4):589–611

Zliobaite I (2014) Controlled permutations for testing adaptive learning models. Knowl Inf Syst 39(3):565–578

Zliobaite I, Bifet A, Read J, Pfahringer B, Holmes G (2015) Evaluation methods and decision theory for classification of streaming data with temporal dependence. Mach Learn 98(3):455–482

Zliobaite I, Kuncheva LI (2009) Determining the training window for small sample size classification with concept drift. In: IEEE international conference on data mining workshops (ICDMW), pp 447–452

Affiliations

Vinicius M. A. Souza[1,2] · **Denis M. dos Reis[1]** · **André G. Maletzke[1]** ·
Gustavo E. A. P. A. Batista[1,3]

Denis M. dos Reis
denismr@usp.br

André G. Maletzke
andregustavo@usp.br

Gustavo E. A. P. Batista
g.batista@unsw.edu.au

[1] Universidade de São Paulo, São Carlos, SP, Brazil

[2] University of New Mexico, Albuquerque, NM, USA

[3] University of New South Wales, Sydney, Australia