Expert Systems With Applications 213 (2023) 118934

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Review

A survey on machine learning for recurring concept drifting data streams

Andrés L. Suárez-Cetrulo [a,b], David Quintana [b,*], Alejandro Cervantes [c]

[a] Ireland's Centre for Applied AI (CeADAR), University College Dublin, D04 V2N9 Dublin, Ireland
[b] Department of Computer Science and Engineering, Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganes, Spain
[c] Escuela Superior de Ingeniería y Tecnología, Universidad Internacional de La Rioja (UNIR), Logroño, Spain

ARTICLE INFO

Keywords:
Regime change
Online machine learning
Data streams
Concept drift
Meta learning

ABSTRACT

The problem of concept drift has gained a lot of attention in recent years. This aspect is key in many domains exhibiting non-stationary as well as cyclic patterns and structural breaks affecting their generative processes. In this survey, we review the relevant literature to deal with regime changes in the behaviour of continuous data streams. The study starts with a general introduction to the field of data stream learning, describing recent works on passive or active mechanisms to adapt or detect concept drifts, frequent challenges in this area, and related performance metrics. Then, different supervised and non-supervised approaches such as online ensembles, meta-learning and model-based clustering that can be used to deal with seasonalities in a data stream are covered. The aim is to point out new research trends and give future research directions on the usage of machine learning techniques for data streams which can help in the event of shifts and recurrences in continuous learning scenarios in near real-time.

1. Introduction

In the last decade, the digitalisation of different industry sectors has accelerated the growth of information to be processed and stored. This evolution is transforming business analysis processes with automated data pipelines and artificial intelligence (AI) models to support decision-making processes. Despite this fact, the industry continues relying on batch techniques for the application of AI as the de-facto standard. Even in continuous scenarios where sequential deep learning models are used, there is a frequent need for retraining strategies at some point in time. Most of these techniques are, in general, unable to deal efficiently with data updates. This limitation is also apparent in domains where a hidden context may influence the predictive model behaviour in unforeseen ways over time. Moreover, many of these techniques are not scalable for a continuous learning setup.

The problem of concept drift (Tsymbal, 2004) has gained a lot of traction in the last years in many domains involving sensors, robotics, system monitoring or anomaly detection, among others. This is also starting to appear in research works over data streams with structural breaks or regime changes. In those, changes in the behaviour of their generative processes result in interleaved periods of temporal dependence and either cyclic or non-stationary patterns to be followed by abrupt changes that modify their overall behaviour (Masegosa et al., 2020; Suárez-Cetrulo, Cervantes, & Quintana, 2019).

Online incremental machine learning (ML) algorithms are scalable for continuous learning scenarios and able to deal with non-stationarities, shifts, and drifts in the data (Elwell & Polikar, 2011). However, stationary scenarios in ML for data streams (namely recurring concepts) are still a subject of study (Alippi, Boracchi, & Roveri, 2013; Gomes, Gaber, Sousa, & Menasalvas, 2014). Even in scenarios where models previously trained may become relevant again, most of the current algorithms need to relearn previous instances, as these are forgotten due to the stability-plasticity dilemma. This need for retraining results in a waste of computational resources, longer training times, and more significant prediction errors while models are not up-to-date.

This issue has traditionally been approached to deal with non-stationary data. Some authors of the literature on ML for data streams have started to consider stationary scenarios in their algorithms (Ditzler, Roveri, Alippi, & Polikar, 2015; Gama, Žliobaitundefined, Bifet, Pechenizkiy, & Bouchachia, 2014; Gomes, Barddal, Enembreck, & Bifet, 2017; Ramírez-Gallego, Krawczyk, García, Woźniak, & Herrera, 2017; Webb, Hyde, Cao, Nguyen, & Petitjean, 2016). A way to deal with these scenarios is by identifying previous behaviours of the data distribution and retrieving learners using the generative process of the data stream as the current data (Abad, Gomes, & Menasalvas, 2015; Ahmadi & Kramer, 2018; Zheng, Li, Hu, & Yu, 2021). This, known as model reuse, is an increasingly popular technique in the literature on online meta-learning for data streams. The aim of this line of research is to design

* Corresponding author.
E-mail addresses: andres.suarez-cetrulo@ucd.ie (A.L. Suárez-Cetrulo), dquintan@inf.uc3m.es (D. Quintana), alejandro.cervantesrovira@unir.net (A. Cervantes).

https://doi.org/10.1016/j.eswa.2022.118934
Received 21 May 2022; Received in revised form 19 September 2022; Accepted 27 September 2022
Available online 3 October 2022

strategies to update, replace and forget learners to suit the data stream dynamics and minimise losses in both accuracy and computational cost.

Data stream learning is a wide field with multiple challenges being addressed by the community. Among them, we could mention multi-label classification (Alberghini, Barbon Junior, & Cano, 2022; Roseberry, Krawczyk, & Cano, 2019; Roseberry, Krawczyk, Djenouri, & Cano, 2021), which is very relevant in text streams. Another example is active learning. In the case of high-frequency data streams, it is likely that not all data can be labelled. Note that this problem is related to the semi-supervised learning paradigm. Therefore, in some circumstances learning methods must first select the proper instances to request labelling and, secondly, use only a small subset of labelled instances. Both topics are challenging and are the subject of recent research (Korycki, Cano, & Krawczyk, 2019; Krawczyk & Cano, 2019). However, due to the increasing broadening of the data stream learning literature, in this work we shall not address these topics in depth, as we intend to focus on techniques to deal with concept changes of different nature.

With this survey, we try to bring attention to how new machine learning techniques for data streams, such as meta-learning approaches, can help in the event of structural breaks in data with temporal dependence and seasonality. We will discuss recurring concept drifts where previously learned models may suit future regimes. Our goal is to point out how to leverage modern ML algorithms that work in continuous scenarios and deal in real-time with any changes that may arise. To do so and provide a self-contained survey, we will describe the problem and types of concept drift and different techniques used in the literature to handle it. These approaches may help find strategies to improve prediction accuracy during times of change, avoid the high computational burden of model retraining, and still benefit from using up-to-date models.

The rest of the document is structured as follows:Section 2 introduces relevant literature on the problem of concept drift. This is important to understand the rest of the paper. Sections 3 to 5 list relevant methods to this survey. Section 3 reviews the supervised learning literature for data streams. Many of these techniques, such as ensemble methods, can be used to store a pool of previously learned models and enhance the predictive accuracy of a model under different seasonalities. Then, Section 4 focuses on meta-learning to deal with recurrences and stationary behaviours in data streams. These techniques combine both supervised and unsupervised learning to detect and react to different concept drifts. Section 5 reviews the unsupervised learning literature to model the state of a data stream. The algorithms that will be covered, not always specific to the data stream learning literature, can help identify clusters identifying different concepts, acting as a trigger for model reuse strategies. Section 6 provides a brief discussion of the importance of the different methods covered in the event of recurring scenarios and compares some of the most important methods from the literature. The final section will be reserved for a summary and to outline future research trends.

2. Concept drifts in continuous learning

Concept drifts can be seen as changes in the data distribution and evolution of relationships between attributes and the target feature over time (see Fig. 1), or as transitions between generative processes in data streams. These can occur at different speeds, severity and distributions.

Drifts can be classified differently depending on the impact, interval, distribution or speed of the change, and can change continuously to novel scenarios or reoccur (e.g. in the presence of cyclic or stationary events). There are various approaches to categorise concept drifts considering these aspects:

- *Impact on the boundaries of the data distribution*. The literature makes a distinction between real and virtual concept drift. The first one affects decision boundaries and deteriorates the performance of the models learned. The second one (virtual) only impacts the conditional probability density function. Virtual drifts, also known as feature or data drifts, do not necessarily impact the posterior probabilities as real drifts do (Ramírez-Gallego et al., 2017). Fig. 2 illustrates a difference between these two types of drift.
- *Distribution and reach of change*. Drifts can occur within a given class or clusters or among many of these (Hu, Kantardzic, & Sethi, 2020), hence, being considered local or global drifts, respectively.
- *The interval of occurrence of drifts*. If drifts always occur at regular time intervals, these are considered periodic drifts. If the time of occurrence is not stable, these are considered irregular.
- *Speed of change*. This speed is defined by the number of instances or batches until the shift completes, and the change is considered completed when data is only generated by the new process (Minku, White, & Yao, 2010). Fig. 3 represents the different types of transitions graphically depending on the speed of change.
  - *Sudden or abrupt drifts* occur in short periods. Usually, in very few data instances. For example, if the generative process of a data stream changes between two consecutive data instances or batches $C_{t_i} \neq C_{t_{i-1}}$, we talk about a sudden drift.
  - *Gradual drifts* are characterised by a more moderate speed than sudden drifts. These exhibit a longer transition phase, and data instances are generated by a mixture of the previous $C_{t_{i-1}}$ and the new C_{t_i} concepts.
  - *Incremental drifts* are characterised by the slowest speed of change, and differences between data instances in the transition period may not even be statistically significant.
- *Recurrence*. Recurring (or recurrent) drifts are transitions to concepts previously seen. These represent a change leading to stationarity in the data stream. For instance, if the data stream has a set of states $DS = \{S_1, S_2, \dots, S_n\}$, where each state S_i is generated by a different generative process C_i, transitions to these known processes (e.g. S_2) are considered recurring drifts when they occur as a new drift S_n. This is formally well defined in Ramírez-Gallego et al. (2017) as $\mathrm{S}_{i+1} = \mathrm{S}_{i-k}$, where k represents the k_{th} previous generative process. Recurrent drifts can be sudden, gradual or incremental depending on their speed, and periodic or irregular depending on their repetition intervals.
- *Blips (or outliers) and noise* tend to be ignored in the literature as they may represent random shifts for short time frames and represent residual concept transitions, respectively.

In the case of recurring drifts, previously learned models may become relevant again in the future. Considering potential concept recurrences is suitable since, as part of the stability-plasticity dilemma, online incremental machine learning algorithms may have to relearn previous concepts if these do not have explicit mechanisms to remember them. This process has a high computational burden, as it implies adapting or training a new model from scratch. Thus, it impacts the predictive accuracy while the models are not up to date with the latest state of the data stream.

Wares, Isaacs, and Elyan (2019) described important challenges encountered in concept drifting data streams. Online ML models need first to learn the latent representation of the dataset, handle changes in the probability distribution and deal with *catastrophic forgetting*. Catastrophic forgetting, a challenge faced in continuous learning and especially in evolving data streams (Korycki & Krawczyk, 2021b; Szadkowski, Drchal, & Faigl, 2021), refers to adaptive learning models forgetting previous knowledge when learning new patterns over time.

An open issue in the data stream learning field is the lack of links with the traditional time series literature (Della Valle, Ziffer,

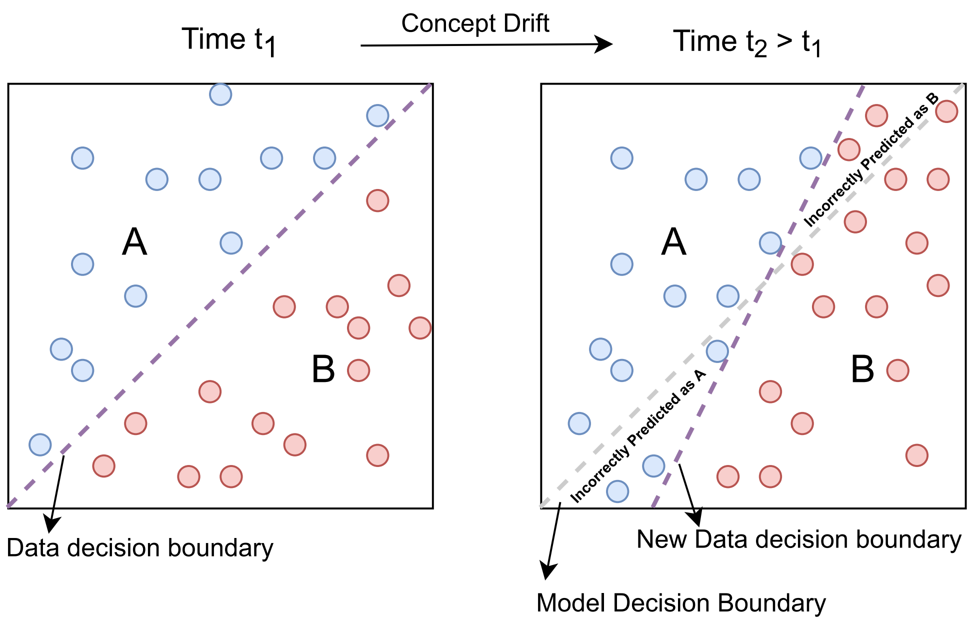
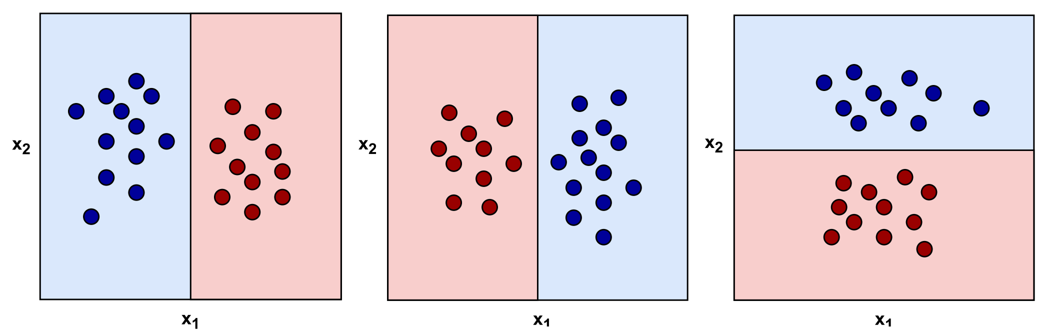


Fig. 1. Illustrative example of a concept drift.



(a) Original data (b) Virtual Drift (c) Real Drift

Fig. 2. Types of drift depending on their areas of influence. x_1 and x_2 represent two attributes of the feature set. Classes in blue and red colours.

Bernardo, Cerqueira, & Bifet, 2022; Gomes, Read, et al., 2019). This has started being studied by Jesse Read recently (Read, 2018; Read, Rios, Nogueira, & De Mello, 2020), who aimed to unify the concepts of data streams and time series by assessing their definitions in the literature and theoretical formulation. This author proposed to interpret concepts as temporal sequences. This allows continuous adaptation and the transfer of knowledge to the next concept as an effective alternative to explicit concept drift detection. He noted that approaches such as stochastic gradient descent would be able to perform this continuous adaptation. He referred to the literature of transfer learning (TL) (Pan & Yang, 2009), and therefore neural networks (either shallow or deep), as a way to deal with concept drifting data streams.

The approach suggested by Read was based on the temporal dependence that he perceived in the context of gradual or partial concept drifts, a behaviour also observed by De Mello, Vaz, Grossi, and Bifet (2019). Although traditionally, TL has been used in offline settings, requiring the entire training set to be present in memory before training commences, a few recent studies applied it to non-stationary data streaming environments (Minku, 2019). Antoñanzas, Arias, and Bifet (2021) presented a way to include information about the current data distribution and its evolution over time into ML algorithms.

Several approaches from the deep learning (DL) field (e.g. RNNs) have also tried to face the problem of concept changes when learning continuously (Korycki & Krawczyk, 2021b; Szadkowski et al., 2021; Zhang, Liu, & Zuo, 2021). Gomes, Read, et al. (2019) analysed the usefulness of DL and reinforcement learning (RL) methods in data streaming applications and further covered the links between time series and data streams. Both techniques work naturally for prediction in streaming contexts. Still, these have not been widely researched by the data stream learning community due to their difficulty in training and their need for reward functions instead of true labels.

2.1. Adaptation to changes

The main difference between incremental and adaptive learning algorithms is that the second group considers explicit strategies to forget irrelevant information. In fact, according to Gama et al. (2014), adaptive learning learners can be interpreted as "*advanced incremental learning algorithms*" that can adapt to changes in a data stream. The evolving nature and speed of dynamic environments in data stream learning trigger a set of issues at the storage and learning stage in ML algorithms. In this domain, the underlying generative process of a time series can change over time; models trained on old data instances may reduce their performance under such changes. Hence, a priority in this field is to create mechanisms to handle and adapt to concept drifts (Lu et al., 2019) while still accounting for periods of stability. Bahri, Bifet, Gama, Gomes, and Maniu (2021) provided a survey discussing research constraints and the state-of-the-art in supervised and non-supervised learning in data stream learning.

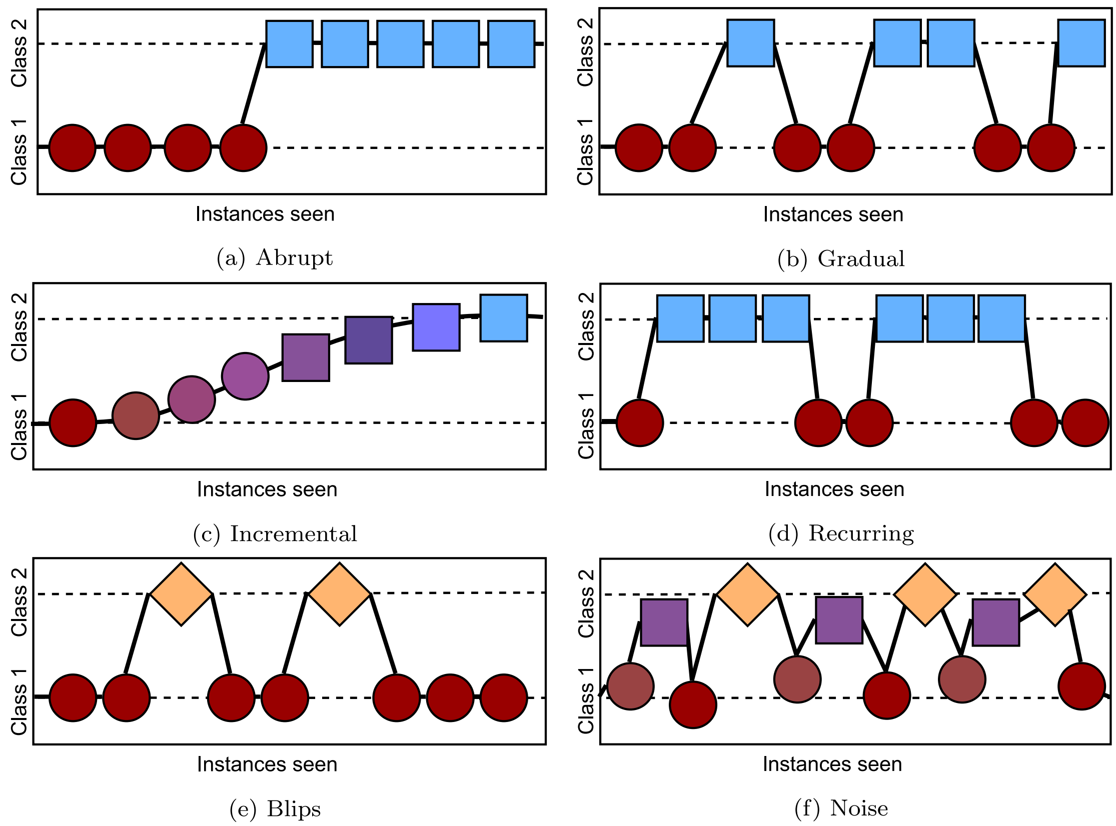


Fig. 3. Different types of drifts depending on their speed and sharpness.

The trade-off between cost efficiency and performance is one of the most significant challenges in data stream learning (Sethi & Kantardzic, 2017; Žliobaitė, Budka, & Stahl, 2015). Online ML algorithms have additional requirements compared to offline learners, such as the need to process instances incrementally to avoid storing data for multiple passes. In terms of passive adaptation approaches to concept drift, ensembles have been one of the mechanisms with the greatest predictive and computational performance. Ensembles naturally fit the purpose of distributed computing frameworks, being easily scalable to deal with massive data streams. However, the incremental nature of the online learning process is purely sequential and becomes a computational bottleneck for base learners running in parallel. In these scenarios, an adaptive framework can benefit from a mini-batch learning strategy as proposed in Cassales, Gomes, Bifet, Pfahringer, and Senger (2021). Mini-batch approaches have been widely used in the literature on data stream learning to port offline state-of-the-art algorithms to work with dynamic environments or non-stationary data.

Continuous model training has become a convention in the data stream learning literature, as it typically focuses on concept-drifting data streams. Along these lines, prequential or interleaved test-then-train has become a popular evaluation mechanism in data stream learning since it helps monitor the error of an algorithm over time (Cerqueira, Torgo, & Mozetič, 2020; Suárez-Cetrulo, Kumar, & Miralles-Pechuán, 2022; Tieppo, Santos, Barddal, & Nievola, 2021). In a prequential evaluation, data is continuously evaluated as soon as it is available. Each data instance is used first to predict. Then, once the ground truth is known, this instance can be labelled, and the prediction error of the model can be computed. Finally, the model can be updated, using that instance for training. The prequential scheme is illustrated in Fig. 4. The main difference between this approach and the convention in machine learning, namely the holdout scheme, is that in the prequential evaluation, test and train are not disjointed sets. The prequential scheme uses instances more efficiently (Cerqueira et al., 2020). It is suitable for online and incremental algorithms that can adapt to drifts, avoiding retraining strategies when dealing with non-stationary data (Žliobaitė et al., 2015).

Continuous learning may only be seen as a requirement in non-stationary scenarios, as other stationary processes could be handled by model reuse. Having said this, in an infinite data stream there may be small non-stationary learnable patterns inside each state, and stationary states could also evolve continuously over time. Žliobaitė et al. (2015) proposed a framework to assess the utility of having adaptive learning learners in different prediction problems. Furthermore, each use case and domain may need different adaptation strategies, but the manual development of a strategy is a time-consuming process. Bakirov, Fay, and Gabrys (2021) proposed a flexible mechanism to automate the development of adaptation strategies. However, this is a very recent proposal; its use is not widespread and does not have enough competitors.

Another research area of interest for classification using online data streams is the one of evolving intelligent systems (EIS) (Angelov & Zhou, 2008; Baruah & Angelov, 2011), or evolving fuzzy systems (Lughofer, 2011; Pratama, Anavatti, Joo, & Lughofer, 2015). These online and incremental systems are able to adapt themselves to concept drifts of different natures on-the-fly through adaptive fuzzy-rules (Angelov & Filev, 2004). These have achieved great results classifying non-stationary time series (Gu, Angelov, Ali, Gruver, & Gaydadjiev, 2016; Pratama, Lu, Lughofer, Zhang, & Er, 2016; Pratama, Lughofer, Er, Anavatti, & Lim, 2017). Recent works in this area, such as Cano and Krawczyk (2019a), aim for GPU parallelism and interpretable models (Angelov & Soares, 2020).

EIS approaches can work as ensembles of rules (Angelov, 2017) and apply meta-cognitive scaffolding theory for tuning the learned model incrementally in what-to-learn, when-to-learn, and how-to-learn (Sateesh Babu, Suresh, & Huang, 2011). These have also introduced the ability to deal with recurrent concepts explicitly. For instance, Pratama

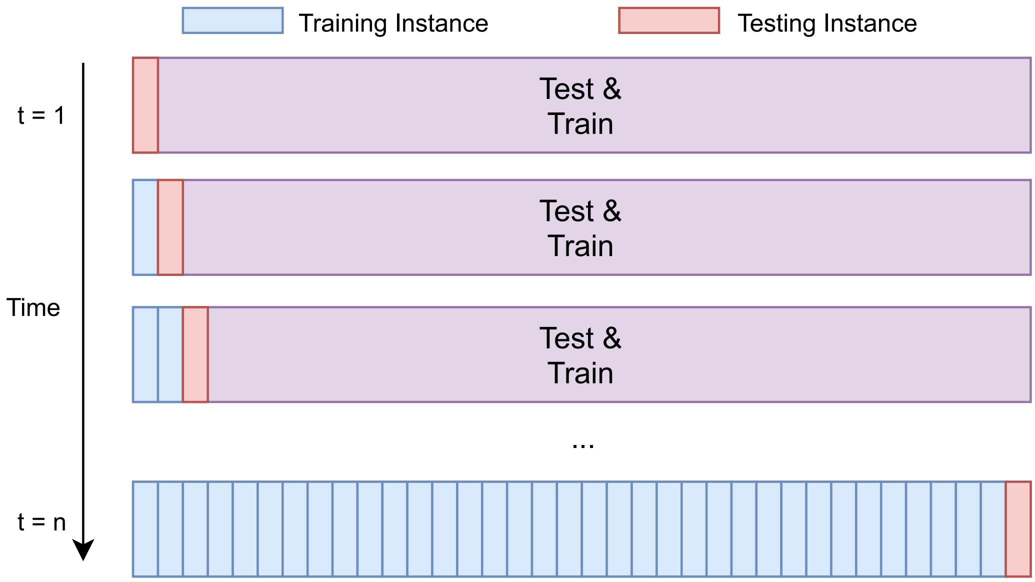


Fig. 4. Prequential scheme. Each instance is used first for test, and then to train.

et al. (2016) employed evolving type-2 recurrent fuzzy neural networks to learn incrementally and handle recurring drifts. In any case, there is still a significant gap between EIS and the rest of the literature on data stream classification.

2.2. Drift detection

While the literature has proved that continuous adaptation is a good mechanism to handle gradual or incremental drifts, incremental learners may need time to adapt in case of abrupt changes due to the model trained already for previous instances of a prior concept. In these cases, the convention is to monitor either change points (or intervals, as done by Basseville, Nikiforov, et al. (1993)) in the data distribution or the predictive performance of a learner to quantify or characterise a drift.

When drifts occur, algorithms in this field tend to entirely or partially replace models once significant changes are detected. Many online classifiers use embedded drift detectors. For instance, HAT (Bifet & Gavaldà, 2009) uses the detector ADWIN2 at each tree node, cutting branches if a drift is detected. These grow again as new data instances from new concepts are learnt.

A set of drift detectors known for state-of-the-art results with different types of data (De Barros & De Carvalho Santos, 2019; Gonçalves, de Carvalho Santos, Barros, & Vieira, 2014) in the data stream learning literature can be found below.

- *ADaptive WINdowing 2* (ADWIN2). This detector proposed by Bifet and Gavaldà (2007) maintains a sliding window divided into two sub-windows representing old and new data and adjusting dynamically. ADWIN2 signals drift if the mean difference between both sub-windows surpasses a threshold. The window size decreases in the presence of drift and increases during periods of stability. This detector has recently been used in the literature to detect concept drift using online classification error rates (Gomes, Bifet, et al., 2017). A separate instance of ADWIN2 needs to be in place with a lower threshold to detect warnings.
- *Drift detection method* (DDM). DDM (Gama, Medas, Castillo, & Rodrigues, 2004) uses classification results to compute the online error rate of the base learner, working under the assumption that when the concept changes, the base learner will incorrectly classify the arriving instances that are created by a different generative process. Thus, if the error rate increases up to a certain threshold, it raises a concept drift signal.
- *Reactive drift detection method* (RDDM). This detector was proposed by De Barros, Cabral, Gonçalves Jr, and Santos (2017) as an improvement of DDM, which sensitivity decreased over time in very large concepts. RDDM continuously recomputes the statistics responsible for signalling warnings and drifts. It discards old instances and forces drift in concepts active for long periods or with long warning windows.
- *Drift detection method based on Hoeffding's inequality* (HDDM). HDDM, by Frias-Blanco et al. (2014), applies "*probability inequalities that assume only independent, univariate and bounded random variables to obtain theoretical guarantees for the detection of such distributional changes*". HDDM monitors false positive and negative rates, not assuming that the results are given by a Bernoulli distribution. The authors of HDDM propose two different versions:
  - *A-test* (HDDM_A) uses two moving averages to track changes.
  - *W-test* (HDDM_W) uses weighted moving averages instead.

  HDDM's A-test and W-test are aimed for abrupt and gradual changes, respectively (De Barros & De Carvalho Santos, 2019).
- *Early drift detection method* (EDDM). This variant of DDM (Baena-Garcıa et al., 2006) analyses the distance between two consecutive misclassifications instead of the number of misclassifications. One advantage of this detector is that it does not have any input parameter.

Lu et al. (2019) reviewed the state-of-the-art in concept drift detection and adaptation, unifying the general framework used by most literature works for this purpose. This, illustrated in Fig. 5, is made of many stages: (i) Retrieval of data stream instances (both historical and new); (ii) Data pre-processing and modelling; (iii) Test statistic calculation and posthoc tests.

The fast growth of research works highlighting the importance and proposing new drift detection mechanisms has recently triggered the publication of several works surveying and benchmarking the main explicit drift detectors. Angelopoulos et al. (2021) and Chikushi, De Barros, da Silva, and Maciel (2021) benchmarked the impact of different detectors across different state-of-the-art incremental classifiers. Gonçalves et al. (2014) and Pesaranghader and Viktor (2016), compared state-of-the-art concept drift detectors. DDM performed well in both works. A summary of the best performers in their study is reported in Table 1.

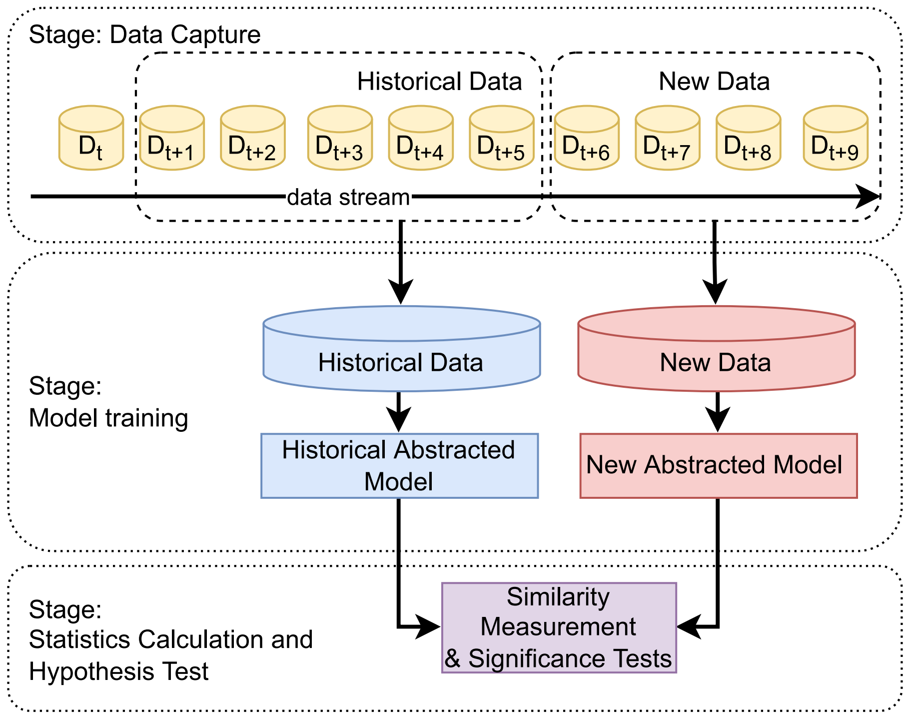


Fig. 5. Example overall framework for concept drift detection.
Source: Adapted from Lu et al. (2019).

Table 1
Extract of the comparison of classification accuracy in data streams with abrupt, gradual drifts and real-world data by Gonçalves et al. (2014). MD: mean of examples seen until drift detection; FA: false alarms; MD: miss detection rates.

Methods	Metrics							
	Abrupt accuracy	Gradual accuracy	Real-world accuracy	Overall accuracy	Execution time	DT	FA	MD
ADWIN2	⇓	⇑	⇑	⇑	⇑	⇓	⇑	⇓
DDM	⇑	⇑	⇓	⇑	⇑	⇑	⇑	⇑
EDDM	⇓	⇓	⇓	⇓	⇑	⇑	⇓	⇓

De Barros et al. (2017), introduced the RDDM drift detector. They then benchmarked different concept drift detectors as auxiliary methods in ensembles in terms of final predictive accuracy under abrupt and gradual concept drifts (De Barros & De Carvalho Santos, 2018, 2019). RDDM and HDDM_A were the best detectors overall, depending on the type of drift (gradual or abrupt) and the base classifier used (naive Bayes or a Hoeffding tree).

Something in common between these studies is that the performance of these detectors over a classifier depends on the nature of the data stream and the base classifier (De Mello et al., 2019). Chikushi et al. (2021) benchmarked detectors across classifiers for different datasets and concluded that every type of data needs its own evaluation process to decide on a suitable drift detector.

2.3. *Drift-related performance metrics*

The replacement or updating of a machine learning model upon concept drift may affect the long-term performance of the model if changes are misdetected. In such scenarios, adding a drift detector may jeopardise the recognition of future recurrences or make models not generalise and overfit specific patterns of the data stream. For these reasons, many research works have proposed metrics to evaluate the performance of concept drift detection methods. Bifet (2017) proposed different metrics to measure false alarms in drift detection such as (i) *mean time between false alarms*; (ii) *missed detection rate* (accounting for the non-detected changes); and (iii) *mean time to detection* (detection delay).

Recently, the literature has shown that, despite the good performance exhibited by many supervised drift detectors, the error rate of the learners that these detectors monitor can be based on temporal dependence (Bifet, 2017; Della Valle et al., 2022; Žliobaitė, Bifet, Read, Pfahringer, & Holmes, 2014). Adaptation errors in these scenarios can occur if this temporal dependence is ignored, leading to reduced predictive performance due to suboptimal decisions regarding what base learners to use for predictions (Halstead et al., 2021). Hence, in cases of sharp changes where passive adaptation is not enough, many recent approaches to handle drifts considering temporal dependence have been proposed (Da Costa, Duarte, Vallim, & De Mello, 2017; Da Costa, Rios, & De Mello, 2016; Vallim & De Mello, 2014).

Indeed, the idea of perfect change detectors lies in the accurate identification of all changes in the generative process of the data stream. In practice, all the studies presented in the previous subsection tend to underestimate or overestimate the number of changes in a stream. The task of concept drift detection does have associated costs, both in predictive accuracy and computational terms. This has encouraged researchers to propose different metrics to measure the accuracy of detection versus the ground truth and the associated costs while a model retrieves from a change.

To handle concept drift in real-world data, detectors face the problem of ignoring when an actual (ground truth) change occurs and for how long. Furthermore, there is no real baseline for the predictive accuracy of a learner if the concept drift does not happen. Hence, it is not feasible to guarantee that a learner has *recovered* from a drift. To do this effectively, the ground truth changes in the data stream must be known.

In many domains, the most reasonable way to know this is through controlled experiments generating synthetic datasets that simulate ground truth changes. This guarantees the existence of real switches, hence allowing the evaluation of change detectors. In this scenario, all changes identified by drift detectors should be counted as *false alarms* if these occur before the real one. Apart from this, two metrics that should be considered would be the number of times a model is

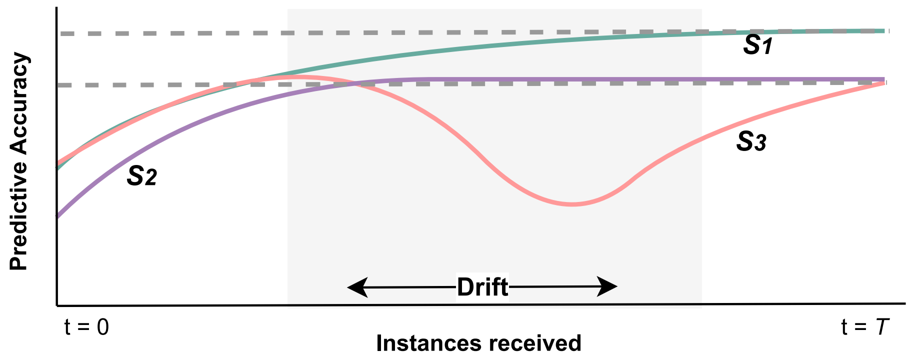


Fig. 6. Schematic illustration of the recovery analysis graphical process.
Source: Adapted from Shaker and Hüllermeier (2015).

replaced accurately and whether this occurred at the right time. The design of concept drift detectors in the data stream learning literature is described as a trade-off between maximising the true changes and minimising both the number of false alarms and the recovery phase of the underlying classifiers.

Drift detectors add extra complexity that is expected to be compensated with faster adaptation and higher accuracy. In this regard, Žliobaitė et al. (2015) analysed how, depending on the scenario or the problem to solve, the cost to detect or react to a concept drift may not be worthwhile. Metrics such as the duration of the recovery phase and maximum performance loss, proposed by Shaker and Hüllermeier (2015), may suit this purpose.

Shaker and Hüllermeier (2014) applied a *survival analysis* method to recognise dynamic events in data streams. Survival analysis (also known as "time to event" analysis) represents a set of statistical tools used to determine when a particular event will occur. Inspired by this work, they introduced *recovery analysis* a year later (Shaker & Hüllermeier, 2015), which is an experimental protocol and a graphical presentation of the learner's performance in a data stream. They also proposed measures to maintain the quality and generalisation performance of the models. Their protocol aimed to estimate the inherent delay in recognising changes and the recovery time; being this last, when will the system recover from an event. Finally, synthetic data has the risk of being idealised and unrealistic. Hence, many approaches try to produce semi-synthetic datasets simulating many data streams and changes in the generative processes over time. Fig. 6 depicts an example of this graphical process. Instead of using a single data stream, their proposal worked with three data streams in parallel: two "*pure streams*" and one "*mixture*" stream.

The three lines in Fig. 6 represent the performance of a model of these three data streams. The darker grey area represents the time window when the concept drift occurs. S_1, S_2 represent the performance of a model trained in two data streams representing two different concepts. S_3 represents the performance of a model trained in a data stream that drifts from concept 1 to concept 2. It can be seen that from the start of the drift, there is a recovery phase until $t = T$ when the performance of the model trained with the concept drifting stream converges with the performance of the model trained only with S_2. The recovery analysis protocol proposes metrics to estimate this recovery's length (t) and the maximum drop in S_3 after a drift. Shaker and Hüllermeier (2015) proposed mainly two metrics to measure drift detection: (i) the *duration* of the recovery phase (suboptimal performance of the algorithm) and (ii) *maximum performance loss* (max. error peak).

Other authors that have proposed metrics in this regard are Žliobaitė et al. (2015), who defend that models should adapt to drifting concepts if the improvement over the error exceeds the cost of the resources required for such adaptation. Hence, they proposed RAM-hour (megabytes of RAM used per hour) as a performance metric to report the runtime cost in prequential learning tasks.

3. Supervised learning under concept drift

Online algorithms should be designed to be single-pass (or onepass learning), always be up-to-date and ready to predict, and handle changes in data streams (Tieppo et al., 2021). Thus, when trying to adapt offline ML algorithms to work in an online setting, a common approach is to fill a buffer with the incoming data instances and train them using in a mini-batch learning setting to support (batch) incremental learning (Gama et al., 2014). An example is *online incremental support vector machines* (OISVM) (Zheng, Shen, Fan, & Zhao, 2013), proposed as a mini-batch learning approach for SVMs.

Other more modern approaches are purely incremental algorithms such as *adaptive random forest* (ARF) (Gomes, Barddal, et al., 2017) and block-based ensembles, such as adaptive *XGBoosting* (AXGB) (Montiel et al., 2020), which are introduced in the subsections that follow.

3.1. Online ensembles

In evolving data streams, changes may be followed by stable periods of different duration. Ensembles can store a set of weak learners trained during different periods, which makes them suitable techniques to adapt to concept drifting data streams (Rokach, 2010). In fact, offline ensembles are known for their good results in predicting both cyclic and non-stationary data (Ballings, Van Den Poel, Hespeels, & Gryp, 2015; Patel, Shah, Thakkar, & Kotecha, 2015a, 2015b). In the last years, many incremental ensembles have been proposed in the data stream learning literature (Krawczyk & Cano, 2018; Krawczyk et al., 2017) to deal not only with stationary data and recurring drifts but also with non-stationary data in evolving data streams (Elwell & Polikar, 2011; Gomes et al., 2014; Hosseini, Ahmadi, & Beigy, 2011, 2012; Karnick, Ahiskali, Muhlbaier, & Polikar, 2008; Kolter & Maloof, 2007; Li, Wu, & Hu, 2012).

Gomes, Barddal, et al. (2017) proposed a taxonomy for data stream ensemble learning derived from reviewing the most relevant approaches at that point in time, covering aspects like the aggregation of predictions, methods to achieve diversity, and type of model updates. According to Gomes, Read, et al. (2019), ensembles for data streams traditionally could be divided into two groups depending on their approach to handling concept drifts:

- *Passive ensembles* (reactive) are updated continuously and assign weights to base models depending on their latest or accumulated predictive accuracy. Some examples of passive ensembles are the *streaming ensemble algorithm* (SEA) (Street & Kim, 2001), *dynamic weighted majority* (DWM) (Kolter & Maloof, 2007) and *accuracy updated ensemble* (AUE) (Brzezinski & Stefanowski, 2014).
- *Active ensembles* apply drift detection algorithms to reset weak learners. An example of these is ADWIN bagging, which combines ADWIN2 (Bifet & Gavaldà, 2009) and online bagging (Oza & Russell, 2001).

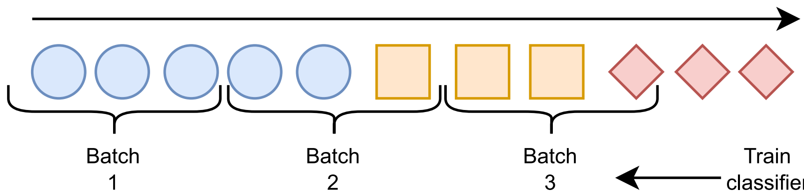


Fig. 7. Example training process of a block-based reactive ensemble.
Source: Adapted from Yang, Wu, and Zhu (2006).

Some more recent works combining both approaches are the adaptations of random forest (RF) and random patches (Louppe & Geurts, 2012) algorithms; *adaptive random forest* (ARF) (Gomes, Bifet, et al., 2017) and *streaming random subspaces* (SRP) (Gomes, Read, & Bifet, 2019; Gomes, Read, Bifet, & Durrant, 2021), respectively. These models can weight different weak learners based on their past performance and replace them when a drift detector specific to their base learners detects a change.

ARF updates its base trees continuously (Hoeffding trees by default). However, if a warning is detected, it starts training a new tree in the background only with new incoming data instances. If a drift is signalled, the background tree becomes the new active tree, and the old one is forgotten. ARF trains its Hoeffding trees (HT) using a resampling algorithm based on online bagging (Oza & Russell, 2001) and uses ADWIN2 as a drift detector, being thus somewhat similar to ADWIN Bagging. It uses boosting to train classifiers iteratively and increases the weight on instances that have been misclassified. Base learners (trees) are weighted using their prequential error. The main difference between the ARF and SRP algorithms is the logic applied to random subspaces in their base learners. While ARF applies them at every base tree independently (locally), SRP ensures a global subspace strategy that increases diversity across weak learners.

Another approach using boosting was proposed by Montiel et al. (2020), who adapted XGBoosting for data stream learning, namely *adaptive XGB* (AXGB). However, their approach was a block-based (mini-batch learning) ensemble, not purely incremental, like ARF and SRP.

3.2. Block-based ensembles

In block-based ensembles, base learners are trained with a batch of data of a fixed length. Most of these approaches create new ensemble members with new batches, setting maximum ensemble sizes and policies to update or replace the current base learners.

AUE, DWM, Learn++.NSE (Elwell & Polikar, 2011) and the *recurring concept drift framework* (RCD) (Gonçalves Jr, Souto, & De Barros, 2013) are examples of some of the first block-based ensembles for data stream learning. RCD (Gonçalves Jr et al., 2013) considers the detection of warning signals before drifts and incorporates the idea of background models that start training in parallel. At the same time, the predictions are performed by a classifier already trained (active or foreground classifier). The addition of background base learners helps online algorithms to shorten the duration of their recovery phase and lower their maximum performance loss. This has been included in many approaches thereafter like ARF (Anderson, Koh, Dobbie, & Bifet, 2019; Gomes, Bifet, et al., 2017).

Block-based ensembles (see Fig. 7) are, in general, passive approaches. Thus, as mentioned in previous sections, this type of adaptation is not suitable to cope with abrupt drifts since they will adapt slowly to those changes, having out-to-date base learners and weights for the global prediction. For this purpose, RCD uses DDM to detect warnings and drifts. AXGB uses ADWIN2 (just like ARF). Block-based ensembles suffer from the main dilemma of other mini-batch learning algorithms introduced in this section. Small blocks can help to react to abrupt drifts, but this has a computational cost and may damage the predictive accuracy of the ensemble in stable periods (Brzezinski & Stefanowski, 2014). Thus, tuning the block size is of vital importance.

3.3. Base learners for recurrences and seasonalities

ARF (Gomes, Bifet, et al., 2017), one of the state-of-the-art methods in data stream learning, incorporates most of the mechanisms described in this subsection. However, a drawback of this approach is that it lacks an explicit mechanism to deal with concept recurrence or seasonalities in a data stream. Many ensemble learners like Learn++.NSE and DWM propose a robust mechanism to deal with recurrent concepts since base learners are not updated after being inserted into the ensemble.

Approaches like ARF constantly train all active base classifiers, which may make base learners evolve and forget the previously learned concept (catastrophic forgetting) before this concept reoccurs. ARF also discards trees when a drift is detected, so these need to be trained from scratch if a concept reoccurs. For this reason, there are authors who suggest using concept history. This feature enables storing cold copies of previously learned base classifiers to be reused if they become relevant again in the future. The idea of a concept history adds extra challenges to identifying what concept is present in the data stream at each time. Different approaches have been proposed for this purpose, like conceptual equivalence and concept similarity (Li et al., 2012; Yang et al., 2006).

- *Conceptual equivalence* was initially proposed by Yang et al. (2006) and assumes that when two classifiers behave similarly predicting during a time window, both describe the same concept.
- *Concept similarity* was initially proposed by Li et al. (2012) to detect recurring drifts in the absence of labelled data. The approach in Li et al. (2012) aimed to recognise similar concepts using Euclidean distances between clusters representing different concepts (namely *concept clusters*).

The concept history is not an item exclusive to online ensembles. It can also be used for single learners that follow a meta-learning approach to change the base learner for different concepts (see Fig. 9).

3.4. Online learning from imbalanced data

Recently authors are starting to consider the fact that in data stream learning, data likely presents class imbalance over any given time window. This is likely true even in cases where the complete data set is not imbalanced.

An important difference from batch classification approaches is that the problem of class imbalance can be represented by a fixed ratio, and the training process can use correction techniques such as oversampling or undersampling all data to facilitate the learning task. However, in a continuous data stream, there will likely be class balance drift; that is, class ratios will not be stationary over time.

In order to tackle this problem, published works consider two additions to the existing balanced approaches: the first is to adjust classifier evaluation using metrics suited for evaluation of performance in the imbalanced case. For instance, in KUE (Cano & Krawczyk, 2019b), the use of Kappa statistic is suggested for classifier selection and updating in an ensemble of classifiers. Other metrics have also been proposed, such as Generalisation Error (Du, Zhang, Gang, Zhang, & Chen, 2021). In Korycki and Krawczyk (2021a), an alternative mechanism for drift

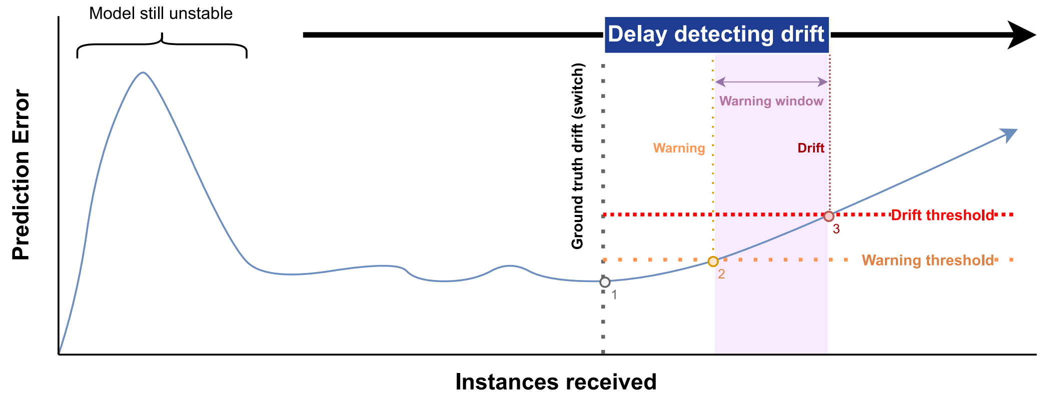


Fig. 8. Illustration of a warning window in a drift detector.
Source: Adapted from Krawczyk et al. (2017).

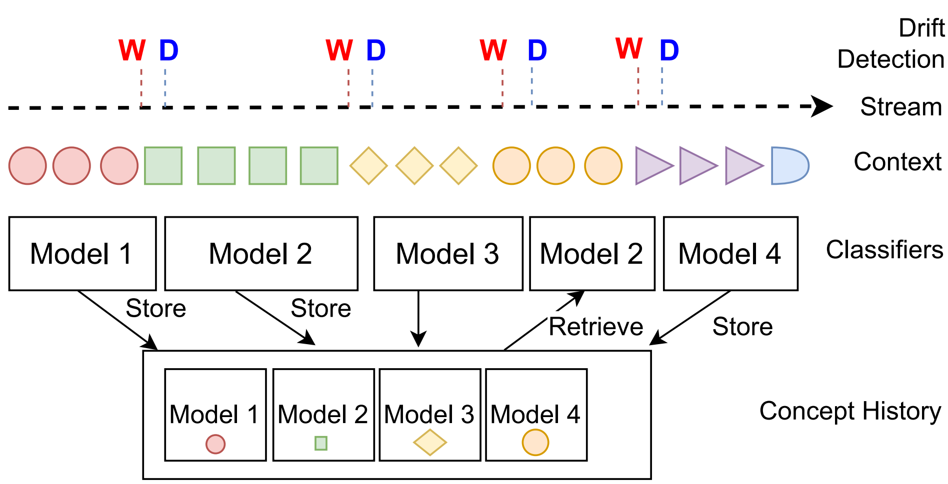


Fig. 9. Example framework using a concept history.
Source: Adapted from Gomes, Menasalvas, and Sousa (2010).

detection is proposed. These authors suggest training a Restricted Boltzmann Machine (RBM-IM) that uses an imbalance-aware loss function to predict change in multi-label online learning problems. This RBM-IM detector is trained in batches and continuously adapted.

Class imbalance poses an additional problem in the form of data sampling. In most circumstances, classifiers benefit from adapting the class distribution to mitigate imbalance. For example, bagging methods can use a modified distribution, sampling for this purpose (Du et al., 2021; Zyblewski, Sabourin, & Woźniak, 2021).

Another approach is used in C-SMOTE (Bernardo et al., 2020) the authors adapt a well-known oversampling method, SMOTE, to oversample the minority class continuously by creating new instances based on available data for the class. C-SMOTE generates the minority class instances required to balance data by monitoring the windows defined by ADWIN. An extensive review of C-SMOTE and other competing resampling methods was published recently (Bernardo & Della Valle, 2022).

In ROSE (Cano & Krawczyk, 2022), resampling is considered when selecting data for training background classifiers with recent data when drift is detected. ROSE can downsample the majority class in the stream by using independent windows to select instances used for each class. This has the effect of providing the novel background classifiers with a balanced data set of recent instances to start their training. The members of this background ensemble compete and eventually replace the former members of the ensemble if they prove more efficient. For replacement, ROSE evaluates the product of accuracy and Kappa score to assess the skill of each classifier properly.

4. Meta-learning and detection of recurrences

Drift detectors collect a series of statistics to signal changes in the underlying data distribution or in the classifier performance over time. This collection process implies a delay between the start of the current drift and the time when this is detected, namely the recovery phase, as seen in Section 2.3. There are two major challenges to reducing the duration and maximum performance loss of the recovery phase: (i) to anticipate when will the next drift occur and (ii) devising what concept will be the next to ensure a faster adaptation (Wu, Koh, Dobbie, & Lacombe, 2021a).

Many approaches try to reduce the impact of the recovery phase by incorporating a warning detector. However, signalling warnings still imply delay from the actual change point since these are based on the same data collection process only using a more sensible parametrisation (e.g. higher confidence intervals in ADWIN2). Models still need to be re-trained from the warning detection point, increasing the computational cost of the training process and not being an effective solution in case of very sharp changes, when a drift will be recognised a few data instances later. In this case, the learner will have a short warning window where the background learners will not be trained with a representative sample of the current concept. In the data stream learning literature, a warning window is known as the time between the warning and drift signals (Abad et al., 2015; Gomes, Menasalvas, & Sousa, 2011). In Fig. 8, the warning window is represented as the purple area that occurs between the detection of a warning (orange dotted line) and the confirmation of a drift (red dotted line) once the threshold of the explicit drift detector is surpassed.

If the data stream presents stationarities or seasonalities at some point in time, a way to alleviate this problem is through model reuse. ML frameworks that do this generally tend to assume that discrete concepts exist. In the last years, many research works have approached the problem of recurring concepts by reusing models trained previously (Yang et al., 2006).

Two methods from the literature that incorporate the reuse of previous base learners are RCD (Gonçalves Jr et al., 2013), introduced in the previous subsection, and the *concept profiling framework* (CPF) (Anderson, Koh, & Dobbie, 2016). These methods, independently of their number of base learners (one or many), act as a wrapper to decide at each time what is the best algorithm to make predictions and allow the use of any base model and detector, being thus meta-learners. The objective of both algorithms is to improve classification accuracy, as for many other relevant methods such as *leveraging bagging* (Bifet, Holmes, & Pfahringer, 2010), and AUE (Brzezinski & Stefanowski, 2014), but this does not necessarily imply improving the detection of drifts.

Many meta-learning approaches aim to represent concepts using non-supervised techniques to describe the current state of the stream and predict potential changes. In this regard, different research works have proposed different mechanisms to represent and measure distances among concepts. Meta-learners generally have more parameters to fine-tune since these may combine different approaches and compare learned models. For this reason, recent research works are starting to propose methods for continuous parameter tuning in non-stationary data streams to cope with this Bahri et al. (2021), Veloso, Gama, Malheiro, and Vinagre (2021).

4.1. Drift detection in meta-learning

Most meta-learning frameworks evaluate models to be reused only when a change is detected. This allows them to leverage offline models, using mini-batch learning approaches and reusing previous models when relevant. For instance, (Trajdos & Kurzynski, 2021) incrementalised an offline classifier using the detector ADWIN2 to signal drifts. In any case, most of the new approaches from the literature work over adaptive learning base classifiers since these can learn gradual changes in the data stream without the help of any explicit drift detection mechanism.

Abad et al. (2015) proposed a meta-learner that uses hidden Markov models (HMM) to predict the sequence of change between discrete concepts. Their approach, which uses fuzzy logic rules to compare classifiers for model reuse, cannot deal efficiently with incoming data streams. Maslov, Pechenizkiy, Žliobaitė, and Kärkkäinen (2016) proposed a method to use patterns acquired during previous changes and assumed a Gaussian distribution for the duration of the changes to predict the time of the next change point.

A similar approach was proposed in Chen, Koh, and Riddle (2016). Their method predicts future changes using a probabilistic network using previous drifts. Their proposal is independent of drift detection methods and relies on volatility patterns in the data stream. ProChange (Koh, Huang, Pearce, & Dobbie, 2018) also used volatility patterns during changes and a probabilistic network to predict different types of drifts in unlabelled transactional data streams. More recently, the authors of Nacre (Wu et al., 2021a), a meta-learner with active drift detection for data streams, proposed a method called drift coordinator to anticipate change points by assessing each concept.

The literature on meta-learning and ensemble learning for data streams is closely related. As mentioned earlier in this section, Gonçalves Jr et al. (2013) proposed the ensemble RCD with active drift detection and a history of previous models to handle recurring concept drifts. The selection of the model to be retrieved is based on the data distribution of the incoming stream. For this purpose, a sample of the data received is stored in a buffer for each classifier and compared to the incoming stream in case of concept drift. Sakthithasan and Pears (2016) applied discrete Fourier transforms to decision trees to capture recurring concept drifts. The evaluation for model reuse was performed by comparing a compressed version of the learned trees.

A recent approach to meta-learning is aimed towards the drift detection component in Yu et al. (2022). Their authors use a pre-trained supervised learning classifier to learn to detect different types of drift over the data stream. The authors suggest a neural network that learns prototypes for each class of drift, including a basic "no drift" class. In this section, we define the word prototype as a set of instances generated or selected by an algorithm. In this work, the network is pre-trained in a supervised way with a dataset with known ground truth. Afterwards, the drift detector can be used as part of any online learning algorithm, with the advantage of being able to predict the occurrence of drift and its type.

4.2. Explicit handling of recurrences

Elwell and Polikar (2011) dealt with recurrent concepts using a block-based reactive ensemble that did not limit the number of base learners. These authors claim that Learn++.NSE trained one concept per batch received. The algorithm employs a weighted voting mechanism using each individual classification accuracy. In the case of recurring concepts, it is expected that the weight of the base learners representing that concept will increase and hence the global prediction will take into consideration old but relevant knowledge.

Conversely, base learners would reduce their weights if their predictive error increases, disregarding their predictions when these do not match the current concept like other similar approaches that came after Almeida, Oliveira, Britto Jr, and Sabourin (2018), Hosseini et al. (2012). The idea of Learn++.NSE is to keep all the learned knowledge in a pool of classifiers, either active (as a part of an ensemble) or inactive (stored as a concept history), to be used in the future when they become relevant. Other related meta-learning ensemble approaches have been proposed to determine a suitable ensemble size (Duda, Jaworski, & Rutkowski, 2017; Pietruczuk, Rutkowski, Jaworski, & Duda, 2016) dynamically.

The use of a *concept history* became popular about a decade ago and has received different names like "pool of classifiers", "concept list" and "concept repository". Authors like Alippi et al. (2013), Gomes et al. (2014, 2010), Li et al. (2012), Yang et al. (2006) proposed the explicit handling of drifts using different techniques to evaluate the relevance of historical concepts. Yang et al. (2006) introduced the idea of a concept repository and the idea of conceptual equivalence in their ensemble classifier RePro. Their method uses a Markov chain to learn concept transitions. Fig. 9 illustrates one of the first approaches targeted to model reuse in data streaming with a flat structure. However, the idea design of a concept history can follow different architectures. For instance, Sidhu and Bhatia (2018) proposed a recurring dynamic-weighted majority (RDWM), alternating two ensembles, one with active learners and another working as a pool of historical models.

Ahmadi and Kramer (2018) presented the GraphPool framework, which maintains a pool of historical concepts and keeps transitions between concepts using a first-order Markov chain to allow model reuse. Like Learn++.NSE, their approach is considered that every batch of data received would represent a new concept, which is not the case in every data streaming application. In the GraphPool framework, similar concepts are merged within the history. Wu et al. proposed the ensemble algorithm PEARL (Wu, Koh, Dobbie, & Lacombe, 2021b) as an extension of ARF that uses a probabilistic graphical model and lossy counting (Manku & Motwani, 2002) for model reuse when a drift is detected. Another meta-learner is CPF, proposed by Anderson et al. (2016). This algorithm has only one active base classifier at a time, handles recurring concepts explicitly using a concept history, and evaluates previous models to be reused when detecting a concept drift. New models are only inserted into the history in case of drift if these do not

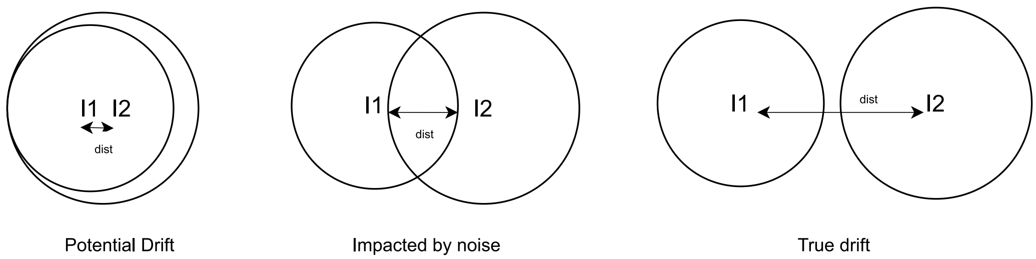


Fig. 10. Cases of concept drift using distances between concept clusters comprehending instances I1 and I2.
Source: Adapted from Li et al. (2012).

perform similarly to any historical one. Historical models are compared using a conceptual equivalence approach, using classification accuracy. To limit and maintain the size concept history over time, CPF prunes historical models using a mechanism called *fading*. It prunes old models depending on how frequently they have been reused, and it gives more importance to older models than recent ones. This design choice may help domains with a finite number of non-evolving concepts. However, it may not suit many real-world data streams in which concepts may evolve. Perhaps due to this, CPF obtained state-of-the-art results in synthetic data with clear recurring patterns but could not outperform other methods, such as RCD, in real-world benchmarks.

One of the limitations of CPF is that it relies on a fixed-size buffer of instances to determine what model to reuse. To avoid the high computational cost of this task and improve its scalability, Anderson et al. (2019) proposed *enhanced CPF* (ECPF). ECPF saves copies of the reused classifiers instead of only the original ones as CPF, which allows concepts to evolve. Furthermore, background learners start being trained when a warning is detected (as in ARF) to replace the active learner in cases where no historical models represent the new concept.

4.3. Concept clusters in meta-learning

Another representative set of online meta-learners is the one focused on an unsupervised representation of concepts and the use of different distance metrics for concept similarity (Abad et al., 2015; Gomes et al., 2010; Li et al., 2012; Menasalvas, Sousa, & Lisboa, 2010; Wu, Li, & Hu, 2012; Zheng et al., 2021). Several of these methods are supervised but have non-supervised concept representation. For instance, the semi-supervised learning tree-based ensemble REDLLA was proposed by Li et al. (2012) for recurring changes in data streaming environments with limited labelled instances. Their approach uses k-means and introduces the idea of concept similarity and *concept clusters*. Fig. 10 illustrates the concept of clusters from this proposal and the idea of a cluster radius to set distance thresholds in the evaluation of similarity.

Li, Wu, He, and Hu (2021) proposed a block-based ensemble model mixing both supervised and non-supervised techniques. Their ensemble has an unlimited-sized concept history and aims at data stream classification. However, it also trains a cluster with every batch received. One of the novelties of their approach was a concept drift detection method based on cluster divergence among batches. Katakis, Tsoumakas, and Vlahavas (2010) presented another reactive block-based ensemble that uses conceptual vectors to represent each batch to approach the idea of a concept history. They used an incremental clustering algorithm to group these vectors and generate concepts. Gomes et al. (2010) and later (Gomes et al., 2014) proposed to use two data streams in parallel. The second data stream relies on user-defined variables specifying context that are then used to proxy the certainty of the ground truth changes. Learners are created over time or reused from a concept repository depending on the similarity of their contextual information. Similar methods were proposed in Abad et al. (2015) and Menasalvas et al. (2010) considering the idea of warning windows (see Fig. 8), later used in ARF and ECPF to train background classifiers. The approach by Gomes et al. (2010) was already illustrated in Fig. 9.

The number of approaches combining unsupervised learning and meta-learning techniques to handle recurring concepts is indeed increasing. Many of these are ensemble learners, which can be purely incremental or block-based. CONDOR (Zhao, Cai, & Zhou, 2020), for instance, is a block-based ensemble. If we compare it to DWM, we identify different weight update strategies and the addition of a meta-learning approach to reuse and update previous models inside the pool of classifiers. Namitha and Santhosh Kumar (2020) presented another cluster-based method to handle recurring concepts in data streams. Their approach is entirely unsupervised, relies on *clustream* (Aggarwal, Philip, Han, & Wang, 2003) and performs unsupervised drift detection.

Sun, Tang, Zhu, and Yao (2018) and Chiu and Minku (2018) proposed ensembles that incorporate concept clusters to limit the ensemble maximum size using a diversity measure. The motivation behind these approaches was that a diverse pool of learners could be more likely to keep a set of representative learners over time with considerably different concepts, which should help in the case of model recurrence. After this, Chiu and Minku (2022) proposed a similar ensemble using Euclidean distances for concept similarity to handle multiple types of drifts. They intended to maximise the diversity of the ensembles having concepts that are distant among them. Their approach, named CDCMS, only creates new models in the ensemble beyond a dissimilarity threshold, and concepts are represented using the *expectation maximisation* (EM) algorithm.

As seen in this section, handling recurring concepts is a topic closely related to ensemble learning for data stream learning and meta-learning. In the latter group, non-supervised learning gains particular relevance to representing concepts and identifying the current state of the ground truth. This offers advantages like bringing the ability to predict changes and facilitate the identification of previous concepts or transition sequences for model reuse. Non-supervised learning is a broader subfield beyond the scope of this section.

5. Model-based clustering under concept drift

The problem of concept drift is a data-stream-learning specific topic, and it involves different challenges in storing, pre-processing and learning from data stream instances. In a non-supervised setting, the number of clusters, their densities, sizes or shapes can evolve due to different non-stationarities in the incoming stream. Recently, Zubaroğlu and Atalay (2021) provided a comprehensive review on data stream clustering algorithms and analysed the non-supervised methods, computational complexity and predictive accuracy of these approaches.

Back in 2001, Wagstaff et al. and two years later (Xing, Ng, Jordan, & Russell, 2003) proposed methods for clustering with similarity information using side-information. Many recent research works have approached the problem of time-changing (and recurring) concept representations in a streaming setting using data stream clustering or deep clustering methods (Din & Shao, 2020; Namitha & Santhosh Kumar, 2020; Zhang et al., 2021). In these, micro-clusters or latent features are used to make a synopsis of the incoming instances and reduce the computational cost of finding similarities among data distributions.

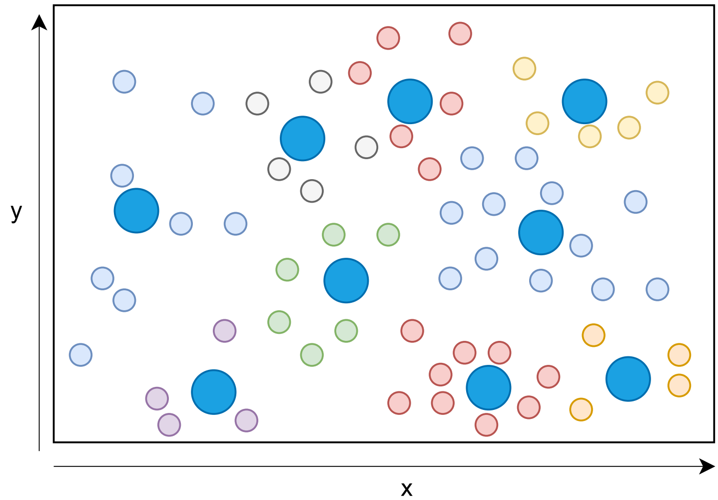


Fig. 11. Data clouds around a set of prototypes (blue dots).
Source: Adapted from Angelov (2017).

However, the problem of representing a concept (or model) using non-supervised learning started becoming popular with model-based clustering approaches (Grün, 2019; McNicholas, 2016). Section 4.3 introduced some research works using the model-based clustering method *expectation maximisation* (EM) (Dellaert, 2002) to improve model reuse. This algorithm, often used in financial applications to identify different market regimes, fits a mixture of Gaussian distributions to the data (Carnein & Trautmann, 2019). Chiu and Minku (2022) used it in CDCMS to create concept representations and keep a diverse ensemble learner. Zheng et al. (2021) used it to minimise the intra-cluster dispersion and cluster impurity. Tsang and Chen (2020) applied the Baum–Welch algorithm, a special case of EM, to both detect the time of a change point and predict the next state (or concept) in time series data using an HMM. Gomes, Read, et al. (2019) also hypothesised about using Baum–Welch in conjunction with HMMs for data streaming scenarios.

In any case, Baum–Welch is not an online approach. Still, it has been used in many domains with other specific versions of EM and Gaussian mixture models (GMM) to forecast changes in generative processes (Park, Lee, Song, & Park, 2009; Zhang, Li, Wang, Fang, & Philip, 2019) and to represent regimes in time series data (Dias, Vermunt, & Ramos, 2015; Kritzman, Page, & Turkington, 2012; Tsang & Chen, 2020). Although incremental versions of the EM and Baum–Welch algorithms have been proposed (Dang, Lee, Ng, Ciptadi, & Ong, 2009; Wakabayashi & Miura, 2009), one of their major disadvantages is their assumption of normally distributed data instances. This can cause many challenges in complex domains or deal with non-stationarity distributions where changes may not be foreseeable.

An alternate approach to represent different concepts is data partitioning. These methods, also unsupervised, can be compared to model-based clustering or other techniques with micro-clusters since all of these can summarise a data distribution into a set of locally optimal structures. For instance, (Angelov, 2017) proposed several ensemble algorithms that have a rule-based model optimised per data cloud (see Fig. 11). They used the term *data cloud* to refer to a set of prototypes identifying a concept and created an autonomous approach to partition data using their data clouds.

As presented in Angelov (2017), the ML literature has many approaches to interpreting the state of data distributions that do not necessarily need to be non-supervised. Among them, we can mention instance-based learning methods such as nearest neighbour-based algorithms. These can be seen as a particular type of prototype-based classifiers (Pekalska, Duin, & Paclík, 2006), which requires the entire dataset to be in memory. A more conventional example of a prototype-based classifier could be the SVM algorithm. Support vectors can be considered prototypes deduced from input data; only these are needed to classify. Something that differs across prototype-based classifiers and impacts the computational cost of the approaches is the selection or generation mechanism to produce prototypes (Angelov, 2017). These can be reduced or generated from the input distribution.

One example of a classifier that generates prototypes is *learning vector quantisation* (LVQ) (Kohonen, 1995). Different incremental algorithms based on this algorithm have been proposed (Xu, Shen, & Zhao, 2012; Zheng et al., 2013). OISVM, presented by Zheng et al. (2013), relies on LVQ to summarise input data and feed it to an SVM classifier to reduce the computational cost at the training stage. This is also a common use of prototype generation techniques like *self-organising maps* (SOM) and *growing neural gas* (GNG) in the literature. Neto and Barreto (2013) provided a good overview of these, and Smith and Alahakoon (2009) compared them by looking at their growth rate, growth conditions, growth inhibition, and data example pruning.

SOM algorithm is an offline method and thus is only able to learn static data. For this reason, it has been adapted in different research works (Liu & Ban, 2015; Prudent & Ennaji, 2005; Si, Lin, & Vuong, 2000) that have introduced dynamic (growing) methods for online learning. An example of these is growing self-organising maps (GSOM) (Fritzke, 1996; Villmann & Bauer, 1998). It grows nodes at the edges of the map when the total distance of an example exceeds a threshold, which allows it to track regions that may present dynamic behaviours when the original SOM would stabilise and lose its capacity to re-shape. GSOM is an incremental approach but not adaptive learning since this does not have a forgetting factor like GNG.

Regarding other surveyed methods to create or select prototypes, SVMs, LVQ or instance-based algorithms (e.g. kNN) are supervised approaches and thus differ from neural gas and self-organising-maps-based techniques. Non-supervised applications like representing non-discrete concepts are not able to leverage these approaches. Moreover, although LVQ has also been used recently to learn in non-stationary environments (Straat, Abadi, Göpfert, Hammer, & Biehl, 2018), to the best of our knowledge, this algorithm, as SOM, has not been widely applied to data stream learning yet. GNG has been proven to effectively reduce the number of instances in a dataset, preserving the original topology (Fritzke, 1995) and has already been used in conjunction with state-of-the-art ML classifiers (Boulbazine, Cabanes, Matei, & Bennani, 2018; Linda & Manic, 2009).

Although these clustering and prototype-based methods are not yet widely used in the data stream learning literature, the research works mentioned in this section, together with recent data stream learning research on concept clusters (see Section 4.3), open good research

Table 2
Design comparison of state of the art classification methods in data stream learning. Abbreviations — DD: drift detector; WW: warning window; Bkg: Background Learner.

	CPF/ECPF	RCD	ARF	RCARF	DWM
DD uses prediction error	✓	✓			(No detectors)
DD reset after drift			✓	✓	(No detectors)
Trains active during WW		✓	✓	✓	(No WW)
Collection of concepts (CH)	✓	✓		✓	
Conceptual equivalence	✓			✓	
Ensemble			✓	✓	✓
Can use any base classifier	✓	✓			✓

prospects for their application modelling concepts. Hence, these can become valuable to detect potential recurrences and identify relevant learners with different approaches other than conceptual equivalence.

6. Discussion

This survey has described the problem of concept drift in data stream learning and covered different strategies that can help during seasonalities in concept drifting scenarios – some of the ensemble methods already discussed, such as Learn++.NSE, DWM, AUC or ARF, among others, can train base learners for a given concept in a specific period –. This approach can help passively handle recurring scenarios since base learners trained for a given recurrence are expected to have greater predictive accuracy. This increases their weight in the overall ensemble and relevance in the global predictions for such periods.

Table 2 covers theoretical differences between many relevant methods from the literature.

Among them, for example, is the error passed for supervised drift detection. ARF uses the error after training with that instance. While the objective of this is not clear in their proposal, we assume that this could work as an approach to flag if a weak learner from ARF does not adapt appropriately to the current concept. Conversely, and according to other relevant ensembles, RCD feeds the prediction error of the active classifier to the detector. Many learners such as CPF, ECPF and RCD do not reset drift detectors at the implementation level after a drift. Hence, allowing warning detection straightway if the error obtained by the base learners is still unstable. Meanwhile, others like ARF and RCARF reset the detectors.

We can see different design strategies to deal with recurrences in the literature. RCARF, RCD, CPF and ECPF keep historical classifiers, although the name given to this concept history changes in each approach. RCD, CPF and ECPF are meta-learners that use any base classifier in their MOA implementations. They also support any drift detector as long as it follows the convention created with DDM, and has a warning zone. While RCD, ARF, RCARF, and DWM are ensemble learners with many active classifiers testing and training at a time, CPF and ECPF only have one active learner in their pool. CPF and ECPF do not train their base learners during the warning window; ARF and RCARF do this but save a snapshot of the classifier at the time of the warning signal; this is inserted in the history in case of concept drift. If the drift is confirmed, the snapshot saved will not include the instances seen during the warning, but the active classifiers are up to date to predict for the time being. DWM, like many other ensemble learners already discussed, does not have a drift detection mechanism.

Anderson et al. (2019) performed a comparison between many of these methods in five different public data streams (see Table 3). ECPF and ARF obtained the best classification accuracies in three and two of the datasets, respectively. Later studies have compared against ARF and ECPF with more than one active classifier. Among them, Wu et al. (2021b) claimed that their online ensemble PEARL beats ARF in streams created with the *Agrawal* data generator (Table 3). Although ECPF obtained greater or similar classification accuracies when using the ensemble ARF as a nested base classifier in many public datasets such as *Electricity* and *Pokerhand*.[1] Gomes et al. (2021) compared ARF, SRP and DWM across 13 data streams, some of which were LED, Agrawal, RBF and Electricity, which are common with the previous works mentioned. They claimed that their proposal, SRS was the classifier with the best classification accuracy across experiments, followed by ARF. Thus, regarding data stream classification, we may conclude that many of these works claim that their approach either beats or compares to the state of the art. Having said that, all of them list ARF among the top performers.

One of the core ideas behind meta-learners like CPF and ECPF is the creation of strategies to replace the active learner over time and reuse previous models if a recurrence is spotted. CPF, ECPF and RCARF use conceptual equivalence. Hence, they consider that a historical learner represents the current concept if it classifies the current data distribution with a lower error than the active classifier. Conversely, several meta-learners like REDDLA and CDCMS represent concepts using non-supervised approaches. The use of concept clusters, which is not as widespread as supervised drift detection, allows learners to track concept drift using alternatives to predictive error. This can help in many ways, such as accounting for virtual drifts or as an extra mechanism to have certainty about changes. In cases of temporal dependence, relying on a single performance metric, especially classification accuracy, can be misleading.

Chiu and Minku (2022) also compared their approach, CDCMS, which represents concepts in a non-supervised manner, in Agrawal, Sea and STAGGER, among other streams. CDCMS obtained the highest prequential accuracy in their study, and RCD was able to obtain comparable results in the streams generated with STAGGER. Along these lines, Li et al. (2021) provided a similar comparison for their unsupervised approach compared to relevant methods such as REDLLA in streams generated with Sea, Waveform and Hyperplane. Their approach, CDMSE, obtained greater predictive accuracy than REDLLA across experiments with lower time consumption.

Finally, in this survey, we have covered relevant works from the machine learning literature that use clustering techniques to represent concepts. While some of these techniques are not part of the data stream learning literature, they are being applied to solve similar problems, like identifying different market regimes in the financial domain. Some of these techniques are based on prototypes (e.g. GNG) and, therefore, might be applicable to concept cluster recognition.

7. Summary and future directions

This paper has reviewed the state-of-the-art in data stream learning and ML to deal with both non-stationary and recurring changes in the generative process of a time series, with a special emphasis on the second group of drifts. The detection of such changes has been traditionally approached using time-series statistical methods, which, as reviewed, still has a gap to bridge with the modern ML literature. Dealing with concept recurrences using techniques such as model reuse is still an emerging subfield of research in data stream learning. With this survey, we hope to have pointed out relevant studies and ways to deal with regime changes in continuous data stream dynamics to support future research in the field.

It is expected that in the near future, techniques such as model reuse and meta-learning frameworks for data stream learning will be thoroughly explored. Especially in the field of MLOps reusing learners previously trained can help save computational costs to have up-to-date models. We can observe a growing trend in the usage of pre-trained models for computer vision, natural language processing and time series supported by communities such as Hugging Face (Wolf et al., 2019). Reusing previous learners can also help in domains such as

[1] https://moa.cms.waikato.ac.nz/datasets/

Table 3
Classification accuracy, Kappa statistic, runtime and memory usage across synthetic datasets as reported in Anderson et al. (2019).

Dataset	Framework	Acc %	σ	Kappa %	σ	Time (s)	σ	Memory (KB)	σ
Agrawal	ECPF	82.2	1.1	64.5	2.2	5.7	0.2	1010	440
Agrawal	CPF	78.2	0.4	56.4	0.8	3.6	0.1	490	100
Agrawal	AUE	74.4	0.1	48.6	0.2	40.4	1.0	180	30
Agrawal	RCD	65.4	2.2	30.7	4.4	34.6	4.1	6570	770
Agrawal	ARF	69.9	0.3	39.5	0.6	226.7	1.8	11,870	4890
CIRCLES	ECPF	93.4	0.1	86.8	0.2	2.1	0.0	190	30
CIRCLES	CPF	93.3	0.3	86.6	0.6	1.4	0.0	370	40
CIRCLES	AUE	94.3	0.0	88.6	0.1	14.0	0.4	110	10
CIRCLES	RCD	90.8	0.2	81.5	0.4	33.3	4.3	720	70
CIRCLES	ARF	94.9	0.0	89.7	0.1	60.2	0.8	710	80
LED	ECPF	72.1	0.1	69.0	0.1	10.6	0.2	270	50
LED	CPF	70.8	0.1	67.5	0.1	7.5	0.1	440	0
LED	AUE	61.0	0.1	56.7	0.1	99.3	3.0	340	20
LED	RCD	65.4	1.3	61.6	1.4	57.3	22.4	2030	150
LED	ARF	61.8	0.1	57.6	0.2	82.1	0.9	640	100
RandomRBF	ECPF	83.1	1.7	65.8	3.3	6.7	0.1	640	270
RandomRBF	CPF	78.1	2.7	55.6	5.3	4.3	0.1	520	140
RandomRBF	AUE	75.6	1.9	50.5	3.5	52.0	1.1	300	430
RandomRBF	RCD	79.6	1.5	58.6	2.8	252.4	24.4	2020	170
RandomRBF	ARF	89.4	1.0	78.5	1.9	128.2	4.5	1890	2320
STAGGER	ECPF	99.7	0.0	99.4	0.0	1.3	0.1	180	20
STAGGER	CPF	98.3	0.0	96.6	0.0	0.7	0.0	180	0
STAGGER	AUE	85.2	0.0	69.9	0.1	8.6	0.3	100	10
STAGGER	RCD	79.0	0.5	56.6	0.9	9.6	1.0	440	30
STAGGER	ARF	99.0	0.0	98.0	0.0	23.5	0.2	320	10

IoT with few-show learning or small data, where the sample is insufficient to train a working model (Wang, Yao, Kwok, & Ni, 2020). Another pending issue in meta-learning and many ensemble learning frameworks is the high computational cost of saving a pool of active or inactive classifiers. More research on forgetting mechanisms to remove old unused models from a concept history would be welcomed.

The majority of the techniques surveyed are focused on data stream classification. There is an increasing trend in porting some of the algorithms reviewed to regression tasks (Lima, Neto, Silva Filho, & Roberta, 2022). However, in the literature on ML for data streams, drift detectors tend to be designed for supervised learning tasks. New methods are yet to be developed to effectively use most online ensembles and meta-learning frameworks for regression tasks. The use of model-based clustering or prototype-based methods opens promising research prospects to model concepts and handle recurrences.

Another research line is the evaluation of concept drift detectors. Although several performance metrics have been proposed for this, the need for a controlled environment where the ground truth regarding concept changes is known limits many real-world applications. In this regard, there is ample room for future research on methodologies to simulate semi-synthetic data in different domains or categorise concept drifts in real-world data streams.

Apart from these, there is much work to be done in areas that fall beyond the scope of this survey, like multi-label classification and concept drift detection with data class imbalance.

Finally, and as many authors have pointed out recently (Gomes, Read, et al., 2019; Read et al., 2020), the data stream learning field needs a more solid research bridge to the time-series literature. Online machine learning algorithms can be seen as predictive and adaptive methods for multivariate data. The problem of model reuse can be interpreted as a transfer learning task. Thus, there is a major opportunity to connect this field with the deep learning literature. This might be the natural evolution of data stream learning to achieve autonomous strategies to decide what-to-learn, when-to-learn and how-to-learn (Angelov & Soares, 2020) as well as automated mechanisms for concept similarity (Chicco, 2021).

CRediT authorship contribution statement

Andrés L. Suárez-Cetrulo: Visualisation, Investigation, Conceptualisation, Methodology, Writing and reviewing. **David Quintana:** Supervision, Conceptualisation, Methodology, Writing and reviewing. **Alejandro Cervantes:** Supervision, Conceptualisation, Methodology, Writing and reviewing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: David Quintana reports financial support was provided by Spanish Ministry of Science, Innovation and Universities.

Data availability

No data was used for the research described in the article.

Acknowledgements

We would like to thank the editor and the anonymous reviewers for their thoughtful and detailed comments on our paper. We would also like to acknowledge the financial support of the Spanish Ministry of Science, Innovation and Universities under grant PGC2018-096849-B-I00 (MCFin).

References

Abad, M. Á., Gomes, J. B., & Menasalvas, E. (2015). Predicting recurring concepts on data-streams by means of a meta-model and a fuzzy similarity function. *Expert Systems with Applications*, *46*, 87–105.

Aggarwal, C. C., Philip, S. Y., Han, J., & Wang, J. (2003). A framework for clustering evolving data streams. In *Proceedings 2003 VLDB conference* (pp. 81–92). Elsevier.

Ahmadi, Z., & Kramer, S. (2018). Modeling recurring concepts in data streams: A graph-based framework. *Knowledge and Information Systems*, *55*(1), 15–44.

Alberghini, G., Barbon Junior, S., & Cano, A. (2022). Adaptive ensemble of self-adjusting nearest neighbor subspaces for multi-label drifting data streams. *Neurocomputing*, *481*, 228–248.

Alippi, C., Boracchi, G., & Roveri, M. (2013). Just-in-time classifiers for recurrent concepts. *IEEE Transactions on Neural Networks and Learning Systems*, *24*(4), 620–634.
Almeida, P. R., Oliveira, L. S., Britto Jr, A. S., & Sabourin, R. (2018). Adapting dynamic classifier selection for concept drift. *Expert Systems with Applications*, *104*, 67–85.
Anderson, R., Koh, Y. S., & Dobbie, G. (2016). CPF: Concept Profiling Framework for recurring drifts in data streams. In *Australasian joint conference on artificial intelligence* (pp. 203–214). Springer.
Anderson, R., Koh, Y. S., Dobbie, G., & Bifet, A. (2019). Recurring concept meta-learning for evolving data streams. *Expert Systems with Applications*, *138*, Article 112832.
Angelopoulos, A., Giannopoulos, A. E., Kapsalis, N. C., Spantideas, S. T., Sarakis, L., Voliotis, S., et al. (2021). Impact of classifiers to drift detection method: A comparison. In *International conference on engineering applications of neural networks* (pp. 399–410). Springer.
Angelov, P. P. (2017). *Empirical approach to machine learning*. Springer.
Angelov, P. P., & Filev, D. P. (2004). An approach to online identification of takagi-sugeno fuzzy models. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, *34*(1), 484–498.
Angelov, P., & Soares, E. (2020). Towards explainable deep neural networks (xDNN). *Neural Networks*, *130*, 185–194.
Angelov, P. P., & Zhou, X. (2008). Evolving fuzzy-rule-based classifiers from data streams. *IEEE Transactions on Fuzzy Systems*, *16*(6), 1462–1475.
Antoñanzas, J., Arias, M., & Bifet, A. (2021). Sketches for time-dependent machine learning. arXiv preprint arXiv:2108.11923.
Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavaldà, R., & Morales-Bueno, R. (2006). Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams, vol. 6* (pp. 77–86).
Bahri, M., Bifet, A., Gama, J., Gomes, H. M., & Maniu, S. (2021). Data stream analysis: Foundations, major tasks and tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *11*(3), Article e1405.
Bakirov, R., Fay, D., & Gabrys, B. (2021). Automated adaptation strategies for stream learning. *Machine Learning*, *110*(6), 1429–1462.
Ballings, M., Van Den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, *42*(20), 7046–7056.
Baruah, R. D., & Angelov, P. (2011). Evolving fuzzy systems for data streams: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(6), 461–476.
Basseville, M., Nikiforov, I. V., et al. (1993). *Detection of abrupt changes: theory and application, vol. 104*. prentice Hall Englewood Cliffs.
Bernardo, A., & Della Valle, E. (2022). An extensive study of C-SMOTE, a continuous synthetic minority oversampling technique for evolving data streams. *Expert Systems with Applications*, *196*, Article 116630.
Bernardo, A., Gomes, H. M., Montiel, J., Pfahringer, B., Bifet, A., & Valle, E. D. (2020). C-SMOTE: Continuous synthetic minority oversampling for evolving data streams. In *2020 IEEE international conference on big data* (pp. 483–492).
Bifet, A. (2017). Classifier concept drift detection and the illusion of progress. In *International conference on artificial intelligence and soft computing* (pp. 715–725). Springer.
Bifet, A., & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining, vol. 7* (pp. 443–448). SIAM.
Bifet, A., & Gavaldà, R. (2009). Adaptive learning from evolving data streams. In *Advances in intelligent data analysis VIII: 8th international symposium on intelligent data analysis, IDA 2009, Lyon, France, August 31 - September 2, 2009. Proceedings* (pp. 249–260). Berlin, Heidelberg: Springer Berlin Heidelberg.
Bifet, A., Holmes, G., & Pfahringer, B. (2010). Leveraging bagging for evolving data streams. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 135–150). Springer.
Boulbazine, S., Cabanes, G., Matei, B., & Bennani, Y. (2018). Online semi-supervised growing neural gas for multi-label data classification. In *2018 international joint conference on neural networks* (pp. 1–8).
Brzezinski, D., & Stefanowski, J. (2014). Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(1), 81–94.
Cano, A., & Krawczyk, B. (2019a). Evolving rule-based classifiers with genetic programming on GPUs for drifting data streams. *Pattern Recognition*, *87*, 248–268.
Cano, A., & Krawczyk, B. (2019b). Kappa updated ensemble for drifting data stream mining. *Machine Learning*, *109*, 175–218.
Cano, A., & Krawczyk, B. (2022). ROSE: Robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams. *Machine Learning*, *111*(7), 2561–2599.
Carnein, M., & Trautmann, H. (2019). Optimizing data stream representation: An extensive survey on stream clustering algorithms. *Business & Information Systems Engineering*, *61*(3), 277–297.
Cassales, G., Gomes, H., Bifet, A., Pfahringer, B., & Senger, H. (2021). Improving the performance of bagging ensembles for data streams through mini-batching. *Information Sciences*, *580*, 260–282.
Cerqueira, V., Torgo, L., & Mozetič, I. (2020). Evaluating time series forecasting models: an empirical study on performance estimation methods. *Machine Learning*, *109*(11), 1997–2028.
Chen, K., Koh, Y. S., & Riddle, P. (2016). Proactive drift detection: Predicting concept drifts in data streams using probabilistic networks. In *2016 international joint conference on neural networks* (pp. 780–787). IEEE.
Chicco, D. (2021). Siamese neural networks: An overview. In H. Cartwright (Ed.), *Artificial neural networks* (pp. 73–94). New York, NY: Springer US.
Chikushi, R. T. M., De Barros, R. S. M., da Silva, M. G. N., & Maciel, B. I. F. (2021). Using spectral entropy and bernoulli map to handle concept drift. *Expert Systems with Applications*, *167*, Article 114114.
Chiu, C. W., & Minku, L. L. (2018). Diversity-based pool of models for dealing with recurring concepts. In *Proceedings of the international joint conference on neural networks, vol. 2018-July*. Institute of Electrical and Electronics Engineers Inc..
Chiu, C. W., & Minku, L. L. (2022). A diversity framework for dealing with multiple types of concept drift based on clustering in the model space. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(3), 1299–1309.
Da Costa, F. G., Duarte, F. S., Vallim, R. M., & De Mello, R. F. (2017). Multidimensional surrogate stability to detect data stream concept drift. *Expert Systems with Applications*, *87*, 15–29.
Da Costa, F. G., Rios, R. A., & De Mello, R. F. (2016). Using dynamical systems tools to detect concept drift in data streams. *Expert Systems with Applications*, *60*, 39–50.
Dang, X. H., Lee, V., Ng, W. K., Ciptadi, A., & Ong, K. L. (2009). An EM-based algorithm for clustering data streams in sliding windows. In *International conference on database systems for advanced applications* (pp. 230–235). Springer.
De Barros, R. S., Cabral, D. R., Gonçalves Jr, P. M., & Santos, S. G. (2017). RDDM: Reactive drift detection method. *Expert Systems with Applications*, *90*, 344–355.
De Barros, R. S. M., & De Carvalho Santos, S. G. T. (2018). A large-scale comparison of concept drift detectors. *Information Sciences*, *451–452*, 348–370.
De Barros, R. S. M., & De Carvalho Santos, S. G. T. (2019). An overview and comprehensive comparison of ensembles for concept drift. *Information Fusion*, *52*, 213–244.
De Mello, R. F., Vaz, Y., Grossi, C. H., & Bifet, A. (2019). On learning guarantees to unsupervised concept drift detection on data streams. *Expert Systems with Applications*, *117*, 90–102.
Della Valle, E., Ziffer, G., Bernardo, A., Cerqueira, V., & Bifet, A. (2022). Towards time-evolving analytics: Online learning for time-dependent evolving data streams. *Data Science*, 16.
Dellaert, F. (2002). *The expectation maximization algorithm*: *Technical report*, Georgia Institute of Technology.
Dias, J. G., Vermunt, J. K., & Ramos, S. (2015). Clustering financial time series: New insights from an extended hidden Markov model. *European Journal of Operational Research*, *243*(3), 852–864.
Din, S. U., & Shao, J. (2020). Exploiting evolving micro-clusters for data stream classification with emerging class detection. *Information Sciences*, *507*, 404–420.
Ditzler, G., Roveri, M., Alippi, C., & Polikar, R. (2015). Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, *10*(4), 12–25.
Du, H., Zhang, Y., Gang, K., Zhang, L., & Chen, Y.-C. (2021). Online ensemble learning algorithm for imbalanced data stream. *Applied Soft Computing*, *107*, Article 107378.
Duda, P., Jaworski, M., & Rutkowski, L. (2017). On ensemble components selection in data streams scenario with reoccurring concept-drift. In *2017 IEEE symposium series on computational intelligence* (pp. 1–7). IEEE.
Elwell, R., & Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, *22*(10), 1517–1531.
Frias-Blanco, I., del Campo-Ávila, J., Ramos-Jimenez, G., Morales-Bueno, R., Ortiz-Diaz, A., & Caballero-Mota, Y. (2014). Online and non-parametric drift detection methods based on Hoeffding's bounds. *IEEE Transactions on Knowledge and Data Engineering*, *27*(3), 810–823.
Fritzke, B. (1995). A growing neural gas network learns topologies. In *Advances in neural information processing systems, vol. 7* (pp. 625–632). MIT Press.
Fritzke, B. (1996). Growing self-organizing networks - Why? In *ESANN'96: European symposium on artificial neural networks* (pp. 61–72). D-Facto Publishers.
Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In *Advances in artificial intelligence – SBIA 2004: 17th Brazilian symposium on artificial intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings* (pp. 286–295). Berlin, Heidelberg: Springer Berlin Heidelberg.
Gama, J., Žliobaitundefined, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, *46*(4).
Gomes, H. M., Barddal, J. P., Enembreck, F., & Bifet, A. (2017). A survey on ensemble learning for data stream classification. *ACM Computing Surveys*, *50*(2), 1–36.
Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., et al. (2017). Adaptive random forests for evolving data stream classification. *Machine Learning*, 1–27.
Gomes, J. B., Gaber, M. M., Sousa, P. A. C., & Menasalvas, E. (2014). Mining recurring concepts in a dynamic feature space. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(1), 95–110.
Gomes, J. B., Menasalvas, E., & Sousa, P. A. C. (2010). Tracking recurrent concepts using context. In *Rough sets and current trends in computing: 7th international conference, RSCTC 2010, Warsaw, Poland, June 28-30,2010. Proceedings* (pp. 168–177). Berlin, Heidelberg: Springer Berlin Heidelberg.
Gomes, J. B., Menasalvas, E., & Sousa, P. A. C. (2011). Learning recurring concepts from data streams with a context-aware ensemble. In *Proceedings of the 2011 ACM symposium on applied computing - SAC '11* (p. 994). New York, New York, USA: ACM Press.

Gomes, H. M., Read, J., & Bifet, A. (2019). Streaming random patches for evolving data stream classification. In *2019 IEEE international conference on data mining* (pp. 240–249). IEEE.
Gomes, H. M., Read, J., Bifet, A., Barddal, J. P., & Gama, J. (2019). Machine learning for streaming data: state of the art, challenges, and opportunities. *ACM SIGKDD Explorations Newsletter*, *21*(2), 6–22.
Gomes, H. M., Read, J., Bifet, A., & Durrant, R. J. (2021). Learning from evolving data streams through ensembles of random patches. *Knowledge and Information Systems*, 1–29.
Gonçalves, P. M., de Carvalho Santos, S. G., Barros, R. S., & Vieira, D. C. (2014). A comparative study on concept drift detectors. *Expert Systems with Applications*, *41*(18), 8144–8156.
Gonçalves Jr, P. M., Souto, R., & De Barros, M. (2013). RCD: A recurring concept drift framework. *Pattern Recognition Letters*, *34*, 1018–1025.
Grün, B. (2019). Model-based clustering. In *Handbook of mixture analysis* (pp. 157–192). Chapman and Hall/CRC.
Gu, X., Angelov, P. P., Ali, A. M., Gruver, W. A., & Gaydadjiev, G. (2016). Online evolving fuzzy rule-based prediction model for high frequency trading financial data stream. In *2016 IEEE conference on evolving and adaptive intelligent systems* (pp. 169–175). IEEE.
Halstead, B., Koh, Y. S., Riddle, P., Pears, R., Pechenizkiy, M., Bifet, A., et al. (2021). Analyzing and repairing concept drift adaptation in data stream classification. *Machine Learning*, 1–35.
Hosseini, M. J., Ahmadi, Z., & Beigy, H. (2011). Pool and accuracy based stream classification: A new ensemble algorithm on data stream classification using recurring concepts detection. In *2011 IEEE 11th international conference on data mining workshops* (pp. 588–595).
Hosseini, M. J., Ahmadi, Z., & Beigy, H. (2012). New management operations on classifiers pool to track recurring concepts. In *Lecture notes in computer science*: *vol. 7448 LNCS*, (pp. 327–339). Springer, Berlin, Heidelberg.
Hu, H., Kantardzic, M., & Sethi, T. S. (2020). No free lunch theorem for concept drift detection in streaming data classification: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *10*(2).
Karnick, M., Ahiskali, M., Muhlbaier, M. D., & Polikar, R. (2008). Learning concept drift in nonstationary environments using an ensemble of classifiers based approach. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 3455–3462).
Katakis, I., Tsoumakas, G., & Vlahavas, I. (2010). Tracking recurring contexts using ensemble classifiers: An application to email filtering. *Knowledge and Information Systems*, *22*(3), 371–391.
Koh, Y. S., Huang, D. T. J., Pearce, C., & Dobbie, G. (2018). Volatility drift prediction for transactional data streams. In *2018 IEEE international conference on data mining* (pp. 1091–1096). IEEE.
Kohonen, T. (1995). Learning vector quantization. In *Self-organizing maps* (pp. 175–189). Springer.
Kolter, J. Z., & Maloof, M. A. (2007). Dynamic weighted majority: An ensemble method for drifting concepts. *Journal of Machine Learning Research*, *8*, 2755–2790.
Korycki, L., Cano, A., & Krawczyk, B. (2019). Active learning with abstaining classifiers for imbalanced drifting data streams. In *2019 IEEE international conference on big data* (pp. 2334–2343).
Korycki, L., & Krawczyk, B. (2021a). Concept drift detection from multi-class imbalanced data streams. In *2021 IEEE 37th international conference on data engineering* (pp. 1068–1079).
Korycki, Ł., & Krawczyk, B. (2021b). Streaming decision trees for lifelong learning. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 502–518). Springer.
Krawczyk, B., & Cano, A. (2018). Online ensemble learning with abstaining classifiers for drifting and noisy data streams. *Applied Soft Computing*, *68*, 677–692.
Krawczyk, B., & Cano, A. (2019). Adaptive ensemble active learning for drifting data stream mining. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 2763–2771). International Joint Conferences on Artificial Intelligence Organization.
Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., Woźniak, M., & Wó Zniak, M. (2017). Ensemble learning for data stream analysis: A survey. *Information Fusion*, *37*, 132–156.
Kritzman, M., Page, S., & Turkington, D. (2012). Regime shifts: Implications for dynamic strategies (corrected). *Financial Analysts Journal*, *68*(3), 22–39.
Li, P., Wu, M., He, J., & Hu, X. (2021). Recurring drift detection and model selection-based ensemble classification for data streams with unlabeled data. *New Generation Computing*, 1–36.
Li, P., Wu, X., & Hu, X. (2012). Mining recurring concept drifts with limited labeled streaming data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *3*(2), 1–32.
Lima, M., Neto, M., Silva Filho, T., & Roberta, A. d. A. (2022). Learning under concept drift for regression-a systematic literature review. *IEEE Access*.
Linda, O., & Manic, M. (2009). GNG-SVM framework - Classifying large datasets with support vector machines using growing neural gas. In *Neural networks, 2009. IJCNN 2009. International joint conference on* (pp. 1820–1826).
Liu, H., & Ban, X.-j. (2015). Clustering by growing incremental self-organizing neural network. *Expert Systems with Applications*, *42*(11), 4965–4981.
Louppe, G., & Geurts, P. (2012). Ensembles on random patches. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 346–361). Springer.
Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, *31*(12), 2346–2363.
Lughofer, E. (2011). *Evolving fuzzy systems-methodologies, advanced concepts and applications, vol. 53*. Springer.
Manku, G. S., & Motwani, R. (2002). Approximate frequency counts over data streams. In *VLDB'02: Proceedings of the 28th International conference on very large databases* (pp. 346–357). Elsevier.
Masegosa, A. R., Martínez, A. M., Ramos-López, D., Langseth, H., Nielsen, T. D., & Salmerón, A. (2020). Analyzing concept drift: A case study in the financial sector. *Intelligent Data Analysis*, *24*(3), 665–688.
Maslov, A., Pechenizkiy, M., Žliobaitė, I., & Kärkkäinen, T. (2016). Modelling recurrent events for improving online change detection. In *Proceedings of the 2016 SIAM international conference on data mining* (pp. 549–557). SIAM.
McNicholas, P. D. (2016). Model-based clustering. *Journal of Classification*, *33*(3), 331–373.
Menasalvas, E., Sousa, P. A. C., & Lisboa, U. N. D. (2010). Tracking recurrent concepts using context in memory-constrained devices. In *Fourth international conference on mobile ubiquitous computing, systems, services and technologies*.
Minku, L. L. (2019). Transfer learning in non-stationary environments. In *Learning from data streams in evolving environments* (pp. 13–37). Springer.
Minku, L. L., White, A. P., & Yao, X. (2010). The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Transactions on Knowledge and Data Engineering*, *22*(5), 730–742.
Montiel, J., Mitchell, R., Frank, E., Pfahringer, B., Abdessalem, T., & Bifet, A. (2020). Adaptive XGBoost for evolving data streams. In *2020 international joint conference on neural networks* (pp. 1–8). IEEE.
Namitha, K., & Santhosh Kumar, G. (2020). Learning in the presence of concept recurrence in data stream clustering. *Journal of Big Data*, *7*(1), 75.
Neto, A. R., & Barreto, G. A. (2013). Opposite maps: Vector quantization algorithms for building reduced-set SVM and LSSVM classifiers. *Neural Processing Letters*, *37*(1), 3–19.
Oza, N. C., & Russell, S. J. (2001). Online bagging and boosting. In *International workshop on artificial intelligence and statistics* (pp. 229–236). PMLR.
Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.
Park, S.-H., Lee, J.-H., Song, J.-W., & Park, T.-S. (2009). Forecasting change directions for financial time series using hidden Markov model. In *International conference on rough sets and knowledge technology* (pp. 184–191). Springer.
Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015a). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259–268.
Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015b). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications*, *42*(4), 2162–2172.
Pekalska, E., Duin, R. P., & Paclík, P. (2006). Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, *39*(2), 189–208, Part Special Issue: Complexity Reduction.
Pesaranghader, A., & Viktor, H. L. (2016). Fast Hoeffding drift detection method for evolving data streams. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 96–111). Springer.
Pietruczuk, L., Rutkowski, L., Jaworski, M., & Duda, P. (2016). A method for automatic adjustment of ensemble size in stream data mining. In *2016 international joint conference on neural networks* (pp. 9–15).
Pratama, M., Anavatti, S. G., Joo, M., & Lughofer, E. D. (2015). pClass: An effective classifier for streaming examples. *IEEE Transactions on Fuzzy Systems*, *23*(2), 369–386.
Pratama, M., Lu, J., Lughofer, E., Zhang, G., & Er, M. J. (2016). Incremental learning of concept drift using evolving type-2 recurrent fuzzy neural network. *IEEE Transactions on Fuzzy Systems*, 1.
Pratama, M., Lughofer, E., Er, J., Anavatti, S., & Lim, C.-P. (2017). Data driven modelling based on recurrent interval-valued metacognitive scaffolding fuzzy neural network. *Neurocomputing*, *262*, 4–27.
Prudent, Y., & Ennaji, A. (2005). An incremental growing neural gas learns topologies. In *Proceedings. 2005 IEEE international joint conference on neural networks, 2005, vol. 2* (pp. 1211–1216). IEEE.
Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M., & Herrera, F. (2017). A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing*, *239*, 39–57.
Read, J. (2018). Concept-drifting data streams are time series; the case for continuous adaptation. arXiv preprint arXiv:1810.02266.
Read, J., Rios, R. A., Nogueira, T., & De Mello, R. F. (2020). *Lecture notes in computer science*: *vol. 12320 LNAI*, (pp. 529–543). Springer, Cham.
Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, *33*(1–2), 1–39.
Roseberry, M., Krawczyk, B., & Cano, A. (2019). Multi-label punitive KNN with self-adjusting memory for drifting data streams. *ACM Transactions on Knowledge Discovery from Data*, *13*(6).

Roseberry, M., Krawczyk, B., Djenouri, Y., & Cano, A. (2021). Self-adjusting k nearest neighbors for continual learning from multi-label drifting data streams. *Neurocomputing*, *442*, 10–25.

Sakthithasan, S., & Pears, R. (2016). Capturing recurring concepts using discrete Fourier transform. *Concurrency Computations: Practice and Experience*, *28*(15), 4013–4035.

Sateesh Babu, G., Suresh, S., & Huang, G.-B. (2011). Meta-cognitive neural network for classification problems in a sequential learning framework. *Neurocomputing*, *81*, 86–96.

Sethi, T. S., & Kantardzic, M. (2017). On the reliable detection of concept drift from streaming unlabeled data. *Expert Systems with Applications*, *82*, 77–99.

Shaker, A., & Hüllermeier, E. (2014). Survival analysis on data streams: Analyzing temporal events in dynamically changing environments. *International Journal of Applied Mathematics and Computer Science*, *24*(1), 199–212.

Shaker, A., & Hüllermeier, E. (2015). Recovery analysis for adaptive learning from non-stationary data streams: Experimental design and case study. *Neurocomputing*, *150*(Part A), 250–264.

Si, J., Lin, S., & Vuong, M.-A. (2000). Dynamic topology representing networks. *Neural Networks*, *13*(6), 617–627.

Sidhu, P., & Bhatia, M. (2018). A novel online ensemble approach to handle concept drifting data streams: Diversified dynamic weighted majority. *International Journal of Machine Learning and Cybernetics*, *9*(1), 37–61.

Smith, T., & Alahakoon, D. (2009). Growing self-organizing map for online continuous clustering. In *Foundations of computational intelligence vol. 4* (pp. 49–83). Springer.

Straat, M., Abadi, F., Göpfert, C., Hammer, B., & Biehl, M. (2018). Statistical mechanics of on-line learning under concept drift. *Entropy*, *20*(10), 775.

Street, W. N., & Kim, Y. (2001). A streaming ensemble algorithm (SEA) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 377–382). ACM.

Suárez-Cetrulo, A. L., Cervantes, A., & Quintana, D. (2019). Incremental market behavior classification in presence of recurring concepts. *Entropy*, *21*(1), 25.

Suárez-Cetrulo, A. L., Kumar, A., & Miralles-Pechuán, L. (2022). Modelling the COVID-19 virus evolution with incremental machine learning. In *The 29th Irish conference on artificial intelligence and cognitive science 2021* (p. 12). CEUR-WS.org.

Sun, Y., Tang, K., Zhu, Z., & Yao, X. (2018). Concept drift adaptation by exploiting historical knowledge. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(10), 4822–4832.

Szadkowski, R., Drchal, J., & Faigl, J. (2021). Continually trained life-long classification. *Neural Computing and Applications*, 1–18.

Tieppo, E., Santos, R. R. d., Barddal, J. P., & Nievola, J. C. (2021). Hierarchical classification of data streams: A systematic literature review. *Artificial Intelligence Review*, 1–40.

Trajdos, P., & Kurzynski, M. (2021). Soft confusion matrix classifier for stream classification. In *International conference on computational science* (pp. 3–17). Springer.

Tsang, E., & Chen, J. (2020). *Detecting regime change in computational finance: data science, machine learning and algorithmic trading.* CRC Press.

Tsymbal, A. (2004). *The Problem of Concept Drift: Definitions and Related Work*: *Technical report: TCD-CS-2004-15*, Department of Computer Science Trinity College, Dublin.

Vallim, R. M., & De Mello, R. F. (2014). Proposal of a new stability concept to detect changes in unsupervised data streams. *Expert Systems with Applications*, *41*(16), 7350–7360.

Veloso, B., Gama, J., Malheiro, B., & Vinagre, J. (2021). Hyperparameter self-tuning for data streams. *Information Fusion*, *76*, 75–86.

Villmann, T., & Bauer, H.-U. (1998). Applications of the growing self-organizing map. *Neurocomputing*, *21*(1–3), 91–100.

Wakabayashi, K., & Miura, T. (2009). Data stream prediction using incremental hidden Markov models. In *International conference on data warehousing and knowledge discovery* (pp. 63–74). Springer.

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM Comput Surv (Csur)*, *53*(3), 1–34.

Wares, S., Isaacs, J., & Elyan, E. (2019). Data stream mining: Methods and challenges for handling concept drift. *SN Appl Sci*, *1*(11), 1–19.

Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., & Petitjean, F. (2016). Characterizing concept drift. *Data Mining and Knowledge Discovery*, *30*(4), 964–994.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

Wu, O., Koh, Y. S., Dobbie, G., & Lacombe, T. (2021a). Nacre: Proactive recurrent concept drift detection in data streams. In *2021 international joint conference on neural networks* (pp. 1–8). IEEE.

Wu, O., Koh, Y. S., Dobbie, G., & Lacombe, T. (2021b). Probabilistic exact adaptive random forest for recurrent concepts in data streams. *International Journal of Data Science and Analytics*, 1–16.

Wu, X., Li, P., & Hu, X. (2012). Learning from concept drifting data streams with unlabeled data. *Neurocomputing*, *92*, 145–155.

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2003). Distance metric learning, with application to clustering with side-information. In *Advances in neural information processing systems*.

Xu, Y., Shen, F., & Zhao, J. (2012). An incremental learning vector quantization algorithm for pattern classification. *Neural Computing and Applications*, *21*(6), 1205–1215.

Yang, Y., Wu, X., & Zhu, X. (2006). Mining in anticipation for concept change: Proactive-reactive prediction in data streams. *Data Mining and Knowledge Discovery*, *13*(3), 261–289.

Yu, H., Zhang, Q., Liu, T., Lu, J., Wen, Y., & Zhang, G. (2022). Meta-ADD: A meta-learning based pre-trained model for concept drift active detection. *Information Sciences*, *608*, 996–1009.

Zhang, X., Li, Y., Wang, S., Fang, B., & Philip, S. Y. (2019). Enhancing stock market prediction with extended coupled hidden Markov model over multi-sourced data. *Knowledge and Information Systems*, *61*(2), 1071–1090.

Zhang, S.-s., Liu, J.-w., & Zuo, X. (2021). Adaptive online incremental learning for evolving data streams. *Applied Soft Computing*, *105*, Article 107255.

Zhao, P., Cai, L. W., & Zhou, Z. H. (2020). Handling concept drift via model reuse. *Machine Learning*, *109*(3), 533–568.

Zheng, X., Li, P., Hu, X., & Yu, K. (2021). Semi-supervised classification on data streams with recurring concept drift and concept evolution. *Knowledge-Based Systems*, *215*, Article 106749.

Zheng, J., Shen, F., Fan, H., & Zhao, J. (2013). An online incremental learning support vector machine for large-scale data. *Neural Computing and Applications*, *22*(5), 1023–1035.

Žliobaitė, I., Bifet, A., Read, J., Pfahringer, B., & Holmes, G. (2014). Evaluation methods and decision theory for classification of streaming data with temporal dependence. *Machine Learning*, *98*(3), 455–482.

Žliobaitė, I., Budka, M., & Stahl, F. (2015). Towards cost-sensitive adaptation: When is it worth updating your predictive model? *Neurocomputing*, *150*(Part A), 240–249.

Zubaroğlu, A., & Atalay, V. (2021). Data stream clustering: A review. *Artificial Intelligence Review*, *54*(2), 1201–1236.

Zyblewski, P., Sabourin, R., & Woźniak, M. (2021). Preprocessed dynamic classifier ensemble selection for highly imbalanced drifted data streams. *Information Fusion*, *66*, 138–154.