

From concept drift to model degradation: An overview on performance-aware drift detectors

Firas Bayram, Bestoun S. Ahmed*, Andreas Kassler

Department of Mathematics and Computer Science, Karlstad University, 651 88 Karlstad, Sweden

ARTICLE INFO

Article history:

Received 11 November 2021

Received in revised form 8 February 2022

Accepted 19 March 2022

Available online 29 March 2022

Keywords:

Concept drift

Model degradation

Data stream

Machine learning

ABSTRACT

The dynamicity of real-world systems poses a significant challenge to deployed predictive machine learning (ML) models. Changes in the system on which the ML model has been trained may lead to performance degradation during the system's life cycle. Recent advances that study non-stationary environments have mainly focused on identifying and addressing such changes caused by a phenomenon called concept drift. Different terms have been used in the literature to refer to the same type of concept drift and the same term for various types. This lack of unified terminology is set out to create confusion on distinguishing between different concept drift variants. In this paper, we start by grouping concept drift types by their mathematical definitions and survey the different terms used in the literature to build a consolidated taxonomy of the field. We also review and classify performance-based concept drift detection methods proposed in the last decade. These methods utilize the predictive model's performance degradation to signal substantial changes in the systems. The classification is outlined in a hierarchical diagram to provide an orderly navigation between the methods. We present a comprehensive analysis of the main attributes and strategies for tracking and evaluating the model's performance in the predictive system. The paper concludes by discussing open research challenges and possible research directions.

© 2022 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In most real-world application scenarios, the machine learning model's performance deteriorates in production and consistently degrades as the systems evolve. This problem is commonly referred to as *model degradation*. The accuracy of machine learning systems is prone to drop for various reasons. One reason could be that the data points on which the model was trained are not sufficient to capture the complexity of the problem space. Therefore, the model will perform unexpectedly for samples in the input space that was not covered in the instance space of training examples [1,2]. Another reason is that the system environment is dynamic and progressively subject to changes, making it difficult for a single model to provide accurate predictions.

In the literature, researchers have distinguished between two main types of system changes concerning their nature. The first type is caused by changes of unknown context that cannot be measured or represented in the available attributes of the dataset, which is known as *hidden context* [3]. Predictive systems typically struggle to cope with changes in hidden contexts, where an adaptive strategy is necessary to be executed. To illustrate the

concept of hidden context, suppose a learning system should predict the Earth's temperature using only spatial and temporal historical data. Over time, the predictions will become inaccurate due to overlooking climate change that serves as a change in the hidden context, which is inaccessible information from the learner's view. Characterizing hidden context is generally domain-dependent, and in most cases, it cannot be expressible in such forms that benefit the learner if it would be incorporated. Therefore, researchers have extensively examined the second type of change, which is diagnosed in the underlying generating function of the data. This phenomenon is known as *concept drift* [4].

Concept drift might be attributed to changes such as degradation in the quality of materials of the system's equipment, seasonality, changing personal preferences and behaviors, or adversarial activities [5]. Since these sources of change are inherent elements of diverse real-world domains, concept drift has been introduced and addressed in a vast range of disciplines and domains. Recent applications include, but are not limited to, IoT systems [6,7], smart grids [8,9], 5G networks [10] and stock market [11,12]. A recent study also has investigated the impact of concept drift on early alert systems during the SARS-CoV-2 pandemic [13]. These explored systems share the non-stationarity property since they are characterized by continuous changes as

* Corresponding author.

E-mail addresses: firas.bayram@kau.se (F. Bayram), bestoun@kau.se (B.S. Ahmed), andreas.kassler@kau.se (A. Kassler).

Acronyms used in the paper**Acronym**

AC	Alternative Classifier
ACCD	Associative Classification over Concept Drifting Data Streams
ACDDM	Accurate Concept Drift Detection Method
ADDM	ADaptive sliding window based Detection Method
ADDS	Anti-concept Drift Detection Algorithm
ADWIN	ADaptive WINdowing
AGE	Accuracy and Growth rate updated Ensemble
ALD	Approximate Linear Dependence
AUE	Accuracy Updated Ensemble
AWE	Accuracy Weighted Ensemble
CDTs	Change Detection Tests
CSDD	Cosine Similarity Drift Detector
CUSUM	CUMulative SUM
DDD	Diversity for Dealing with Drifts
DDM	Drift Detection Method
DDM-OCI	Drift Detection Method for Online Class Imbalance
DELM	Dynamic extreme learning machine
DOED	Diversified Online Ensembles Detection
DWM	Dynamic Weighted Majority
ECDD	EWMA for Concept Drift Detection
ECHO	Efficient Concept Drift and Concept Evolution Handling over Stream Data
ECPF	Enhanced Concept Profiling Framework
EDDM	Early Drift Detection Method
EDIST	Error DISTance for drift detection and monitoring
ELM	Extreme learning machine
ESOS-ELM	Ensemble of online sequential extreme learning machine
EWAUC	Equal Weighted AUC
EWMA	Exponentially Weighted Moving Average
FDR	False Discovery
FHDDM	Fast Hoeffding Drift Detection Method
FHDDMS	Stacking Fast Hoeffding Drift Detection Method
FHDDMSadd	Additive Stacking Fast Hoeffding Drift Detection Method
fnr	False negative rate
FPDD	Fisher Proportions Drift Detector
fpr	False positive rate
FSDD	Fisher Square Drift Detector
FsNB	Fast switch Naive Bayes model

FTDD	Fisher Test Drift Detector
FTRL	Follow the Regularized Leader
FTRL-ADP	Follow-the-Regularized-Leader with Adaptive Decaying Proximal
HDDM	Hoeffding Drift Detection Method
HDWM	Heterogeneous Dynamic Weighted Majority
HLFR	Hierarchical Linear Four Rates
HT	Hoeffding Tree
KME	Knowledge-maximized ensemble
KS	Kolmogorov–Smirnov test
LFR	Linear Four Rates
MD3	Margin Density Drift Detection
MDDM	McDiarmid Drift Detection Method
meta-RRKOS-ELM-DDM	Meta-cognitive Recurrent Recursive Kernel Online Sequential Extreme Learning
MOS-ELM	Meta-cognitive online sequential extreme learning machine
NB	Naive Bayes
NDE	Number and Distance of Errors
NSE	Non stationary environments
OAUE	Online Accuracy Updated Ensemble
ODKK	Online drift detector for a K-class problem
OMR-DDM	Online Map-Reduce Drift Detection Method
OS-ELM	Online sequential extreme learning machine
OWE	On-line Weighted Ensemble
PAUC	Prequential Multi-Class AUC
PHT	Page–Hinkley test
PINE	Predictive and Parameter INsensitive Ensemble
PPV	Positive Predictive Value
PSO	Particle Swarm Optimization
RDDM	Reactive Drift Detection Method
RDWM	Recurring Dynamic Weighted Majority
SEA	Streaming Ensemble Algorithm
SPC	Statistical Process Control
STEPD	Statistical test of equal proportions
SVM	Support Vector Machines
TDAP	Time Decaying Adaptive Prediction
tnr	True negative rate
tpr	True positive rate
UCEM	Uncertainty Error Correlation Matrix
WAC	Weighted AUC
WELM	Weighted extreme learning machine
WMA	Weighted Majority Algorithm
WSTD	Wilcoxon Rank Sum Test Drift Detector

they develop. A broad overview of concept drift applications can be found in [14].

In the last decades, *learning in non-stationary environments* [15] has been intensively studied. Researchers pointed out the necessity to integrate a model degradation detector in the overall learning framework. After deployment, the detector evaluates and tracks the system's performance to control such degradation

in prediction accuracy. The error rate's degradation level is then used to signal concept drift alerts in the system.

Numerous terms and multiple mathematical definitions can be found in the literature to describe the same concept drift type. This lack of unified terminology in the field makes it challenging for researchers to find the correct definition of a given concept drift type. This paper investigates the terms and definitions used to describe the various types of concept drift. We also present a rigorous summary of concept drift and use a novel

hierarchical classification. We categorize the existing approaches that explicitly rely on monitoring the error rate of the base learner to detect concept drift in the system. Incorporating such detection components in the system boosts the robustness of machine learning systems against changes and helps prevent the performance degradation of predictive models in our constantly changing world.

This paper is designed to address the following research questions:

RQ1: What are the different terms that are used in the literature to describe the same type of concept drift, and what are the same mathematical definitions used to describe the same concept drift type?

RQ2: What are the performance-based drift detection methods that were proposed in the last decade, and how can they be represented through a hierarchical classification?

RQ3: How is the model's predictive performance validated and used to track and detect concept drift, and what are the most common techniques utilized in the reviewed methods?

For **RQ1**, we delve into the literature to obtain the various terms that are used to refer to each concept drift type. For **RQ2** and **RQ3**, we survey the proposed performance-based concept drift detection methods in the last decade. From 2011 up to 2020, as illustrated later, researchers have principally extended or got inspired from the benchmark methods proposed in the preceding decade. The extended methods have primarily focused on improving the benchmark methods or enhancing their capability in dealing with more complex problems that involve fast-moving volumes of big data streams. We have chosen to review the works in the last decade since they can be viewed as the representatives of the most recent methods developed and compile the latest research gaps in the field.

The rest of this paper is organized as follows. Section 2 provides general background on drift detection and the related work. Section 3 presents the search strategy that was implemented to address the research questions. Section 4 introduces the terms and definitions that are used in concept drift handling frameworks. Section 5 categorizes the performance-based concept drift detectors and reviews the existing approaches in the literature. An in-depth analysis and discussion on the surveyed methods are presented in Section 6. In Section 7, we conclude the paper by presenting the main finding of this study and identifying future research directions.

2. Background and related work

Drift detection or change detection refers to the methodology that helps determine and identify a time instant or time interval when a change arises in the properties of the target object [16]. This definition has been extended to impose time constraints on the detection delay to enable the learner to adapt to the change efficiently to ensure high-performance [17].

Concept drift detection is a component of the concept drift handling framework that activates the *concept drift adaptation* component, which reacts to the change in the data stream [18]. Subsequently, the system will update the prior knowledge and adjust the learning models to react to the changes properly. This update usually raises the conflicting problem known as *stability-plasticity dilemma* [19]. Here, stability means maintaining the relevant and possibly reoccurring knowledge. At the same time, plasticity describes replacing outdated knowledge in response to the new experience. Ideally, the concept drift solution should achieve a balance between stability and plasticity [20]. Such concept drift adaptation strategy is referred to as *informed adaptation*, or *active approach*, which is triggered upon drift

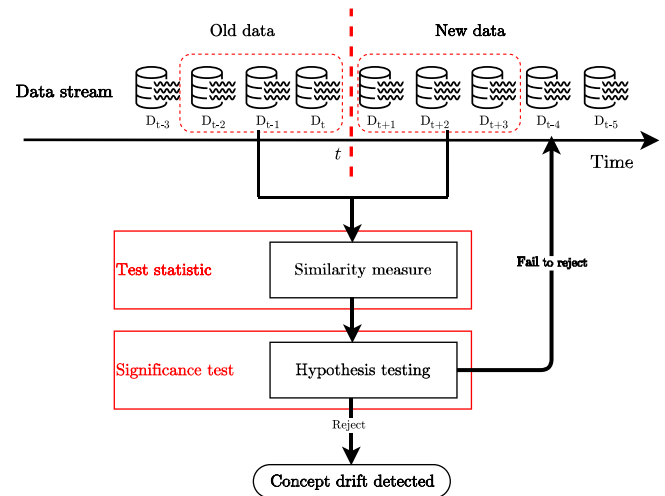


Fig. 1. Concept drift detection framework.

occurrence detection to update the model. The other strategy is *blind adaptation*, also denoted as *passive approach*, where the model is constantly updated upon receiving new data instances without detecting drifts [21,22].

Concept drift detection methods generally use a test statistic to keep tabs on the data stream and quantify the similarity between the old samples and the new ones to discern the change in the concept. This similarity value is then compared with a pre-defined threshold to find out the drift magnitude [23]. Inspired by [18], Fig. 1 summarizes a generic scheme for concept drift detection methods. In the figure, the null hypothesis is that the test statistic will not yield a significant difference between the old and new data, i.e., no concept drift detected. If failing to reject the null hypothesis, the system will persist with the current learner and slide on the data stream.

Existing studies on detecting concept drift can be classified into different categories concerning the test statistics they apply to check and locate the change (see Fig. 2). Data distribution-based and performance-based, or error rate-based, approaches are the most dominant techniques used to detect concept drift since they can be applied to most learning tasks with lower complexity. There are also hybrid and contextual-based approaches.

Data distribution-based detectors use distance measures to estimate the similarity between the data distributions in two different time-windows [24]. Concept drift is then detected if the two distributions are significantly distant. Goldenberg and Webb [25] summarize the distance measures that are used to compare the data distributions and estimate drifts. The main advantage of this approach is that it can be applied to both labeled and unlabeled datasets since this method only considers the distribution of data points. However, as we will discuss later, changes in the data distributions do not always affect the predictor performance, potentially leading to false alarms in the system [26].

Performance-based approaches (as illustrated by the red arrows in Fig. 2) comprise the largest group of concept drift detectors. Therefore, they are the main focus of this paper for surveying and classification. These approaches typically trace deviations in the online learner's output error, known as the predictive sequential (prequential) error [27], to detect changes [28]. The basic idea of performance-based approaches aligns with *Probability Approximately Correct* (PAC) learning model [29], which articulates that the prediction error depends on the size of the examples and the complexity of the hypothesis space. It concludes that if the examples are drawn from a stationary distribution, the

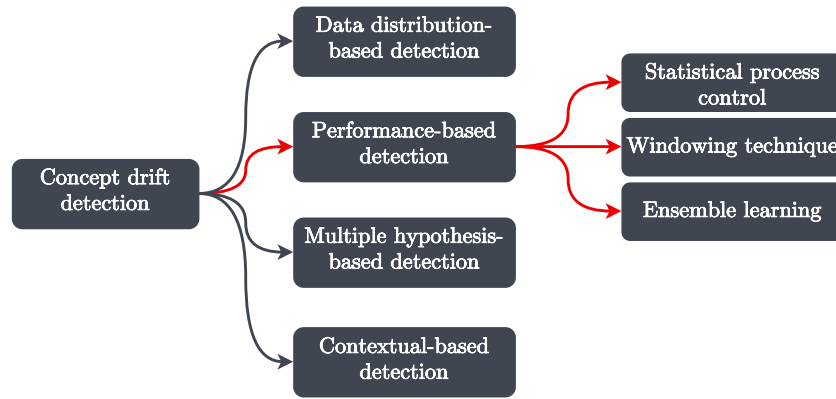


Fig. 2. Concept drift detection methods.

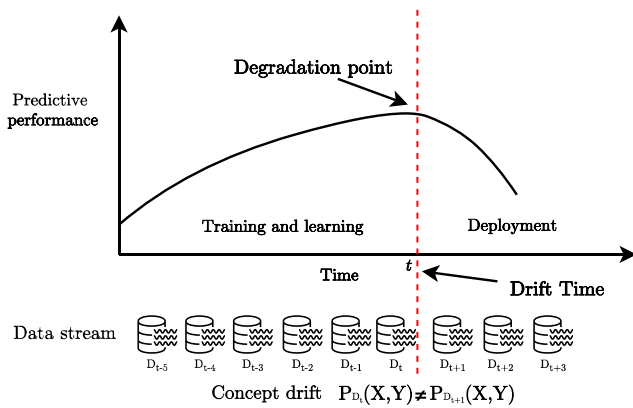


Fig. 3. Performance-based approach mechanism.

error rate decreases as the learner sees more examples [30]. Thus, such a consequential decrease in the performance implies that the learned relationship between the examples of input data and the concept under study is obsolete, resulting in concept drift. Fig. 3 illustrates the main idea of performance-based approach mechanisms. Concept drift occurs when the joint distribution of the dataset $P_D(X, Y)$ changes at time instance t , which is the drift time. The main advantage of performance-based approaches is that they only handle the change when the performance is affected. Thus, these methods are more efficient in dealing with potential false alarms than distribution-based algorithms. However, the main challenge is that these methods require a quick arrival of feedback on the predictions, which is not always available [26]. Because of the limitation mentioned above, a new family of methods that deal with concept drift detection in unsupervised settings has been proposed.

Multiple hypothesis-based drift detectors are hybrid approaches that apply several detection methods and aggregate their results in parallel or hierarchically [18]. Parallel drift detectors integrate the decisions of multiple drift detectors to make the final judgment. Hierarchical drift detectors incorporate two layers for drift detection. The first layer is the warning layer to alert the system about a potential occurrence of concept drift. The second layer is the validation layer that confirms or rejects the warning signaled from the first layer.

Contextual-based detectors use context information available from the system and data to detect the drift. Lu et al. [31] have introduced concept drift detectors in a case-based reasoning system by tracking changes in competence measurement. Demšar and Bosnić [32] have used model explanation methodologies to

interpret, visualize and detect concept drift. Lobo et al. [33] have presented the eSNN-DD method that detects concept drift by exploiting the evolution of spiking neural networks. Huang et al. [34] have designed a concept drift detector using historical drift trends to calculate the probability of expecting a drift using online and predictive approaches. Graph metrics have also been utilized to detect concept drift in data streams that could be represented as graph streams as in [35–37].

Several survey papers to formalize and classify concept drift were presented in the literature. In 2014, the most referenced survey on concept drift was published by Gama et al. [26]. It covered and categorized concept drift handling systems from different perspectives and provided an excellent introduction to adaptive learning and concept drift. Another review paper [18] summarized the research advancements on concept drift and proposed a new component in the concept drift handling framework, called *concept drift understanding*. Ditzler et al. [15] surveyed the studies on concept drift approaches from two main aspects, active and passive. Other related review studies [38–41] have also surveyed and categorized the existing concept drift handling approaches. They have also provided an insightful discussion on the methods. Besides these review papers in the literature, other papers explored handling concept drift in specific learning tasks. A recently published review paper by Gemaque et al. [42] provides a full-scale overview of the methods that handle concept drift in unsupervised learning. Other papers review and scrutinize the progress in class-imbalanced data streams [4,43]. Krawczyk et al. [44] focuses on analyzing the research in ensemble learning for data streams in dynamic environments. However, with the availability of detailed review papers in concept drift classification and formalization, only a few studies have explored the different terms used by authors to describe the same type of concept drift, whereas new terms have appeared since the date of the publication of these studies. Additionally, far too little attention has been paid to performance-based concept drift detection. To this end, one of the main contributions of this paper is to narrow down the focus on performance-based concept drift detection approaches and provide a comprehensive summary of the recent progress in this research area, where not many review studies are available.

3. Search methodology

The main objective of this paper is two-fold. First, the paper formalizes the problem of concept drift and surveys the terminologies used in the literature to describe its types, which is cataloged in **RQ1**. Second, it reviews the recent studies and trends in performance-based concept drift detection, which is addressed in **RQ2** and **RQ3**. Since the field has started to materialize in the

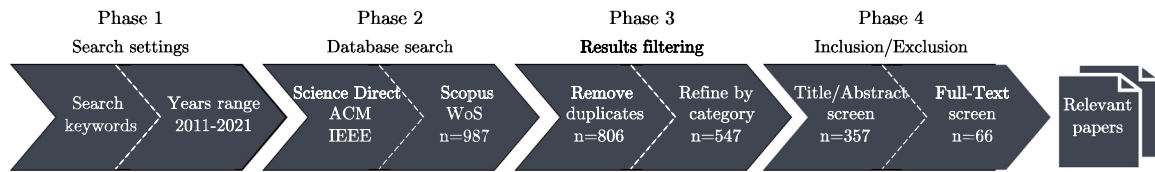


Fig. 4. Search strategy implemented to retrieve the relevant papers in the literature.

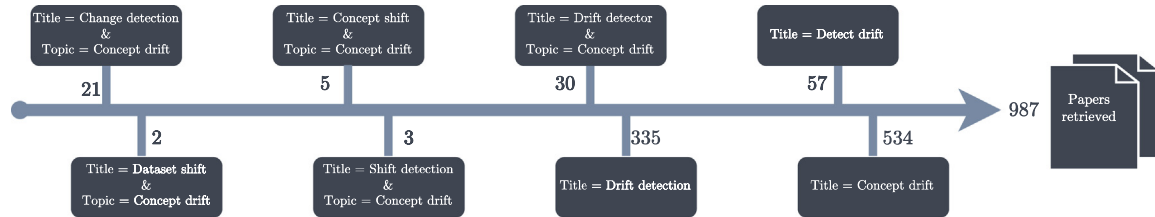


Fig. 5. Search terms for retrieving the papers.

early 2000s, we decided to retrieve the terms that have appeared in the studies of the last two decades while we retrieved the performance-based detection methods of the last decade. The main reason why we have followed different methodologies in addressing **RQ1** and **RQ2** and **RQ3**, is that most of the terms have appeared in the early aughts of the present century and have been used afterward by the authors. In contrast, we limited the retrieval of the detection methods to the last decade to determine the current trends in the research area. To address **RQ1**, we have explored the terms used in former surveys and through snowballing of highly-cited references. For **RQ2** and **RQ3**, and while this paper does not directly follow a systematic literature review protocol, we have followed a systematic literature search methodology to retrieve and select relevant papers that answer **RQ2** and **RQ3**, as shown in Fig. 4. In the first phase, we define the search settings. We set the date range to the last decade (2011–2021) and defined the keywords to use for our search queries since concept drift appeared under different terminologies in the literature. We have summarized the search terms that we used in Fig. 5. In phase 2, we set the paper index database resources we searched to retrieve the papers. We acquired the papers from the top database indices, including IEEE Xplore, Science Direct, ACM, Scopus, and Web of Science. This resulted in 987 publications. In Phase 3, we filtered the results by removing duplicates, resulting in 806 papers. In phase 4, we refine the search results and constrain them to the studies published in the computer science discipline. In addition to the search phases, we performed screening to segregate the relevant papers. We carry out the screening in two levels. The first level investigates the title and abstract to exclude the papers that do not detect concept drift. Then, we scrutinize the full text in the second level to include the relevant studies that use the model's performance to detect concept drift.

To decide whether to include or exclude the paper, we have considered a set of inclusion/exclusion criteria to determine relevant publications.

1. We removed papers that are not published in English.
2. The paper must be peer-reviewed according to the formal peer-review process in the scientific community. That filters out preprints, book chapters, Master or Ph.D. dissertations.
3. Survey papers were excluded, since they do not introduce a new concept drift detection approach.

To determine the approaches relevant to the scope of this paper, we selected the candidate papers according to the following inclusion criteria:

1. The approach must propose a novel drift detection method or integrate existing drift detectors in new predictive systems.
2. The approach must be general and not only targeted to solve a specific problem or installed in a particular domain or application.
3. The approach must explicitly detect concept drift.
4. The approach must use the learner's performance to detect the drift without the underlying data distribution.

Following the above criteria, 66 papers remained to be reviewed for addressing **RQ2** and **RQ3**. The following sections present the result of our analysis and address the research questions.

4. Terminology and definitions

As mentioned previously, researchers have defined and mathematically represented concept drift and its derivatives in different ways. In 2012, Moreno-Torres et al. [45] first addressed this lack of standard terminology and suggested that concept drift is a type of the generic phenomenon *dataset shift* that covers *covariate shift*, *prior probability shift* and *concept drift*. Each concept drift type is framed by a certain change in the data distribution. But since the date of this publication, new terms have appeared in the literature, and novel concept drift types have emerged.

To address **RQ1**, this section will provide a taxonomy to group the various terms used in the literature, starting from mathematical definitions of each variant. The taxonomy will help the researchers and practitioners to gain a unified and consolidated view on the notations by providing precise and concise terminology in the field. In the following subsections, we will formally define concept drift and the different types and survey the terms used to describe each type.

4.1. Notation and formalism for concept drift

In supervised machine learning tasks, each data instance is defined by a pair of feature vectors or covariates X , and a target variable or response y . [26] have introduced a probabilistic definition to describe a time-varying *concept* as the joint distribution of X and y at time t , $P_t(X, y)$. Tracking a change in data samples requires a time-ordered sequence of instances. Concept drift is usually aligned in the *stream learning* context since a data stream is defined as a continuous, potentially unbounded, sequence of data elements with associated time stamps arriving in sequential

order [46]. This is in contrast to dataset shift in a *batch learning* scenario, where the data is entirely stored in memory and processed all at once [40]. Changes are characterized between the training and testing probability distributions [47]. Thus, concept drift is viewed as the stream learning correspondent of the dataset shift in the batch setting [48].

Concept drift is formally defined as a change in the joint distribution between two time instances t and $t + w$ [49], where t could be a particular time point or time interval, and w denotes the time window when the distribution change is being checked at. Consequently, concept drift occurs, if:

$$P_t(X, y) \neq P_{t+w}(X, y) \quad (1)$$

In a recent study [50], authors suggested adding an extra constraint to the definition presented in Eq. (1) to guarantee that the new concept will retain for some time period (at least for two time points):

$$\forall i, \quad w = \tau_{d(i+1)} - \tau_{d(i)} > 1 \quad (2)$$

where $\tau \in \mathbb{Z}^+$ is the time point, and $d^{(i)}$ denotes the time point order of the i th concept drift appeared in the system. This additional constraint will distinguish concept drift from outliers that last momentarily and ensures that the concept drift is a new pattern rather than an ephemeral disturbance in the data (i.e., noise).

Starting from the product rule, and according to the Bayesian Decision Theory [51] the joint distribution in Eq. (1) can be decomposed and rewritten as:

$$P_t(X, y) = P_t(y|X) \times P_t(X) = P_t(X|y) \times P_t(y) \quad (3)$$

In the settings of classification problems,

- $P_t(y|X)$ denotes the posterior probability distribution of the target labels,
- $P_t(X)$ is the input data probability distribution,
- $P_t(y)$ denotes the prior probability distribution of the target labels,
- $P_t(X|y)$ denotes the class-conditional probability density distribution.

4.2. Concept drift types

Researchers categorized concept drift into different types in terms of the form that it takes place in the system. The probabilistic source of change and the arrival pattern (i.e., drift transition) are the most commonly used principles to distinguish concept drift. There are also other criteria to categorize concept drift, such as speed, severity, and recurrence. [39,48] present an exhaustive categorization of concept drift types. The following subsections provide a comprehensive taxonomy of concept drift types, categorized by the probabilistic source of change and drift transition.

4.2.1. Probabilistic source of change

This type of concept drift is the most closely studied in the literature. To make the inequality of Eq. (1) hold, it identifies the changes in the probability distributions. As can be seen from Eq. (3), any concept drift type, assuming probability distribution change, is associated with at least another type since a change in any probability distribution in Eq. (3) will induce at least one change in another distribution. To illustrate that argument, we consider the posterior probability distribution as an example. If $P_t(y|X) \neq P_{t+w}(y|X)$ then, by applying the Bayesian rule:

$$\frac{P_t(X|y) \times P_t(y)}{P_t(X)} \neq \frac{P_{t+w}(X|y) \times P_{t+w}(y)}{P_{t+w}(X)} \quad (4)$$

The inequality of Eq. (4) holds if, at least, one of the probability distributions that compose it has changed. A similar argument can be used for the other probability distributions. The probabilistic sources of drift are then defined as follows:

1. $P_t(y|X) \neq P_{t+w}(y|X)$: A change in the posterior probability distribution indicates a principal change in the underlying target concept. This drift type directly affects the prediction performance since it requires an adaptation of the decision boundary to react to it for preserving the model's accuracy. There are mainly two types, where this form of drift takes place. The first type is mainly referred to as *real concept drift*, Fig. 6(a), where changes in $P(y|X)$ might or might not be associated with changes in $P(X)$ [26]. The second type manifests itself without a change in the data distribution $P(X)$. This type is called *actual drift* [18], as illustrated in Fig. 6(b). This paper mainly covers real concept drift detectors since these methods detect drifts that affect the predictor's performance.

There are also subcategories derived from this probabilistic source of change. *Fickle concept drift* occurs when some data samples belong to two different classes at two different times [52], which can be written mathematically as $\exists x(\text{argmax} P_t(y|x) = c_1 \text{ and } \text{argmax} P_{t+w}(y|x) = c_2)$, Fig. 6(c). *Severe concept drift* occurs if the target classes of all the data samples change after the drift occurrence [53]. This type of drift is also called *full-concept drift* [48], Fig. 6(d). This type of concept drift can be represented mathematically as $\forall x(\text{argmax} P_t(y|x) = c_1 \text{ and } \text{argmax} P_{t+w}(y|x) = c_2)$. *Intersected concept drift* occurs when only a subspace of the data samples changes their target classes after the drift occurrence [53], which is also referred to as *subconcept drift* [48], Fig. 6(e). This type of concept drift can be represented mathematically as $\exists x(\text{argmax} P_t(y|x) = c_1 \text{ and } \text{argmax} P_{t+w}(y|x) = c_2)$ and $\exists z(\text{argmax} P_t(y|z) = c_2 \text{ and } \text{argmax} P_{t+w}(y|z) = c_1)$.

2. $P_t(X) \neq P_{t+w}(X)$: A change in the underlying data distribution is mainly referred to as *covariate shift* [54]. Fig. 6(a) illustrates the covariate shift. If the input data distribution changes without affecting the target concept, and hence the decision boundary, it is called *virtual drift* [55], in mathematical terms, $P_t(y | x) = P_{t+w}(y | x)$ and $P_t(x) \neq P_{t+w}(x)$, Fig. 6(f). In practice, changes in the data and the posterior probability distributions often happen simultaneously [56]. *Local concept drift* and *Feature-evolution* are other subcategories that can be considered as of this probabilistic source of change. *Local concept drift* refers to the situation where the distribution change targets only a sub-region of the feature space [57]. This can be expressed as $P_t(X_1) \neq P_{t+w}(X_1)$ and $P_t(X_2) = P_{t+w}(X_2)$, Fig. 6(g) illustrates the Local concept drift. *Feature-evolution* occurs when new attributes (e.g. X_3) dynamically arise in the input space [58], i.e. when $X_t \neq X_{t+w}$, and as a result, $P_t(X) \neq P_{t+w}(X)$, where X is the set of input variables, Fig. 6(h).

3. $P_t(y) \neq P_{t+w}(y)$ A change of the distribution of classes over time is referred to as *prior-probability shift* [47] as illustrated in Fig. 6(a). This drift type could affect the prediction performance if there is a significant change in the distribution of classes or the number of classes in the learning problem has changed.

Another subcategory of this source of change that is found in the literature is *concept-evolution*, which refers to the emergence of novel classes in the problem [58] as illustrated in Fig. 7(a). Similarly, *concept deletion* refers to the disappearance of classes in the problem [20], Fig. 7(b).

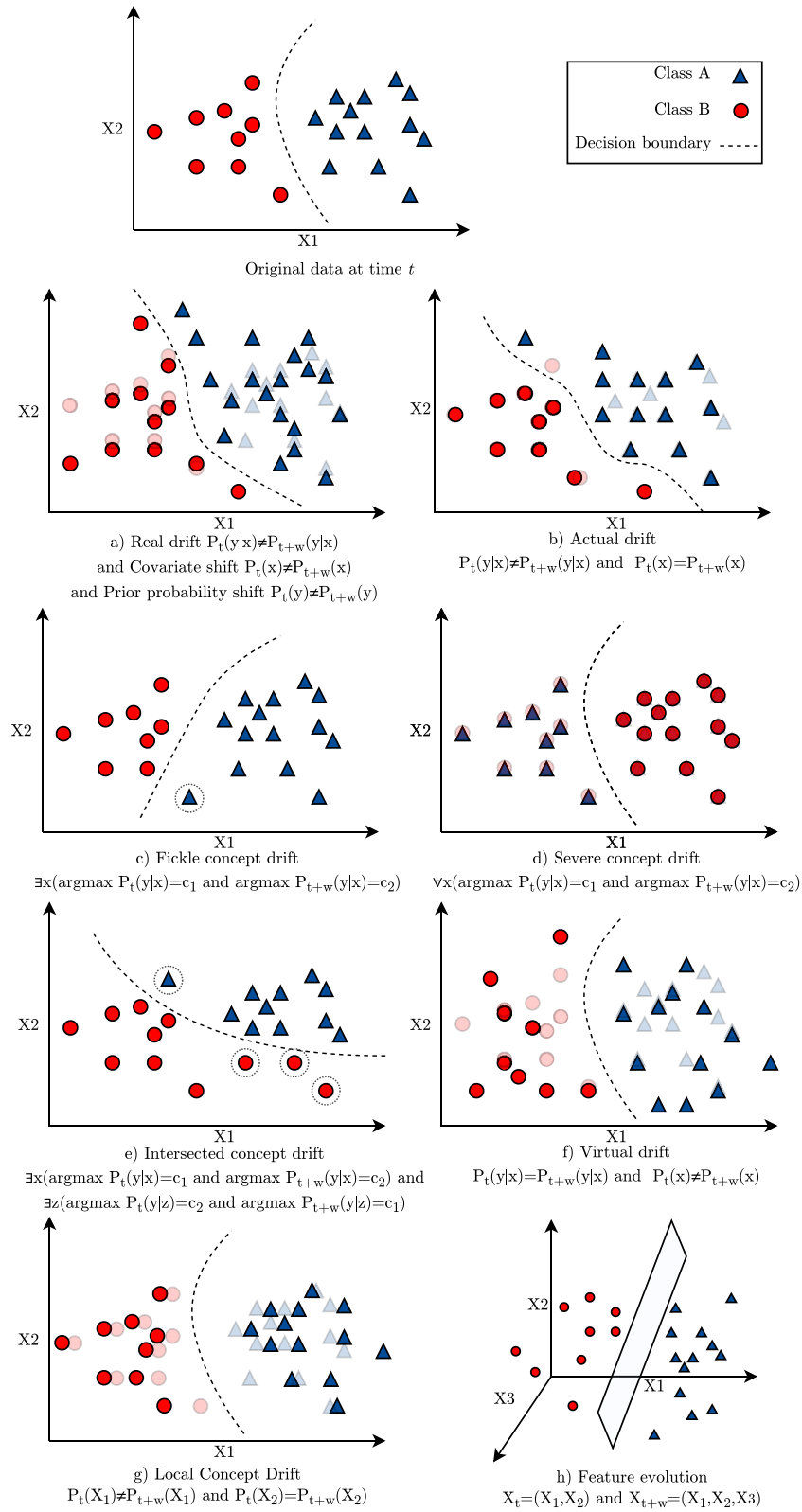


Fig. 6. Concept drift types by probabilistic source of change.

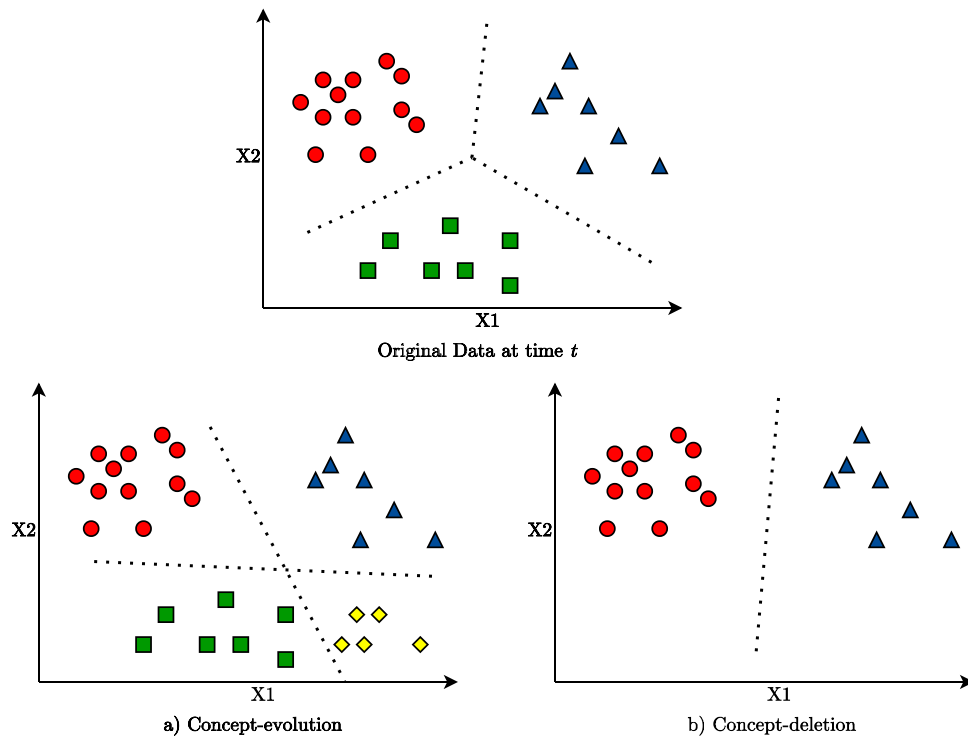


Fig. 7. Concept-evolution and concept-deletion.

Table 1
Concept drift by probabilistic source of change.

Mathematical definition	$P_t(X, y) \neq P_{t+w}(X, y)$	$P_t(y X) \neq P_{t+w}(y X)$	$P_t(y X) = P_{t+w}(y X)$ and $P_t(X) \neq P_{t+w}(X)$	$P_t(y X) \neq P_{t+w}(y X)$ and $P_t(X) = P_{t+w}(X)$	$P_t(X) \neq P_{t+w}(X)$	$P_t(y) \neq P_{t+w}(y)$
Terminologies used	Concept drift [26,59,60]	Real concept drift [26,61,62]	Virtual drift [63,64]	Actual drift [18,65]	Covariate shift [54,66]	Prior-probability shift [45,47]
		Concept drift [20,67]	Temporary drift [68]	Conditional change [69]		
		Concept shift [26,70]	Sampling shift [64,70]			
	Dataset shift [45,47]	Permanent drift [46,68]	Feature change [69]	Real concept drift [41]	Virtual drift [57]	Global drift [71] Label shift [72,73]
		Conditional shift [74,75] Rigorous concept drifting [59]	Loose concept drifting [59]	Concept shift [45,76]	Data distribution drift [77]	Target shift [74,78]
	Concept shift [45,79]	Population drift [80] Class distribution drift [77]	Pure covariate drift [48]	Pure class drift [48]		Class prior shift [4,81]

Table 1 summarizes the terms that can be found in the literature to describe concept drift types that are characterized by the probabilistic source of change. The terms in the table are also grouped by the corresponding mathematical definitions.

4.2.2. Transition of change

This categorization distinguishes concept drift characteristics based on the pattern of how the drift evolves in the system. It can be classified as follows [18]:

1. **Sudden Drift:** Occurs when the target distribution changes from one concept to another abruptly at a point in time (e.g., Fig. 8(a)).
2. **Gradual Drift:** Occurs when the target distribution changes progressively from one concept to another (e.g., Fig. 8(b)).
3. **Recurring Drift:** Occurs when a precedently-seen concept reappears again after a time interval (e.g., Fig. 8(c)). This type is similar to the gradual drift since the two concepts interchange in the system, but the main difference is the transition phase. In gradual drift, the old concept starts to

phase out and is to be replaced with the new one increasingly. While in recurring drift, the old concepts reoccur after some time [55].

4. **Incremental Drift:** Occurs when a new concept replaces the old one slowly in a continuous manner (e.g., Fig. 8(d)). Some Authors consider this type as a sub-type of gradual drift [44], as in the two drift types, the new concepts emerge in the system and completely replace the old one. While the difference is that in the incremental drift, there is no obvious boundary that separates the occurrence of the different concept [82].

Table 2 summarizes the terms that can be found in the literature to describe the aforementioned concept drift types that are characterized by the transition of change.

Tables 1 and 2 answer RQ1 by providing an overview of the different terms used by researchers to refer to concept drift types in the literature.

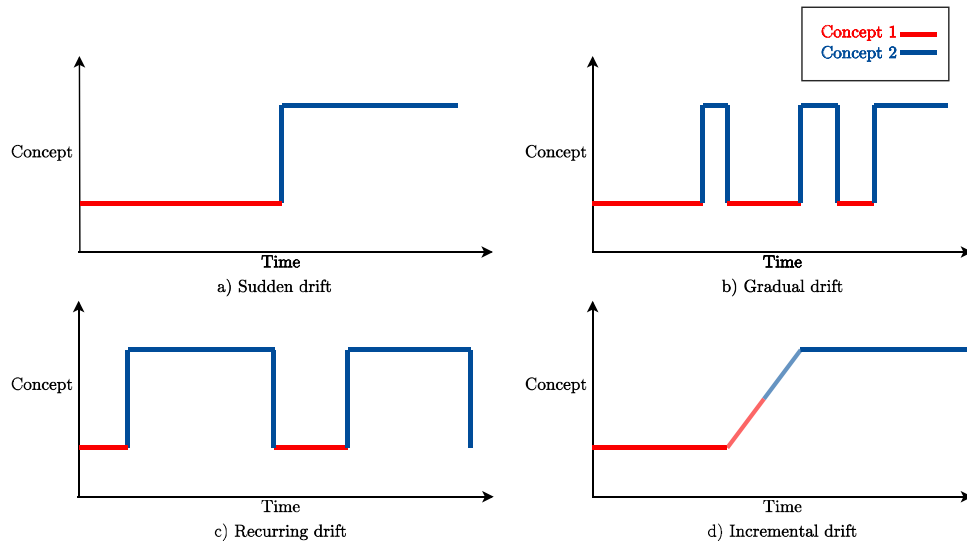


Fig. 8. Concept drift categorized by pattern of arrival.

Table 2
Concept drift by probabilistic source of change.

Primary term	Sudden drift [64,83]	Gradual drift [83,84]	Recurring drift [85,86]	Incremental drift [86,87]
Alternative terms	Abrupt drift [64,88]	Evolutionary drift [89,90]	Recurring contexts [85,91]	Stepwise drift [87,92]
	Concept shift [62,79]		Replacing drift [93]	Development drift [93]
	Revolutionary drift [89,90]			
	Immediate drift [93]			

5. Performance-based concept drift detectors

This section surveys performance-based concept drift detection methods to answer **RQ2**. These methods can be categorized according to the strategy used to detect drops in performance: statistical process control, windowing techniques, and ensemble learning, as illustrated in Fig. 2. To guide the reader, we summarize the reviewed approaches in this paper as illustrated in Fig. 9. The navigation diagram is based on a hierarchical scheme that connects the original method with its derivatives and extensions.

5.1. Statistical process control

The Statistical Process Control (SPC) criterion is used to monitor the quality of the learning process by tracing the online error rate evolution of base learners. Concept drift is assumed to have occurred if the model's performance degradation exceeds the significance test level. Numerous performance-based methods can be found in the literature that rely on SPC to detect concept drift.

The Drift Detection Method (DDM) [30] is a well-known and widely-used algorithm and has been used as conceptual underpinning for a number of related performance-based drift detectors. DDM analyzes the error rate of the streaming data classifier to detect changes. The method considers the error as a Bernoulli random variable with Binomial distribution. It monitors p_t , the probability of misclassification at time t , and the standard deviation s_t as:

$$s_t = \sqrt{p_t(1 - p_t)/i} \quad (5)$$

At time t , p_{min} and s_{min} are replaced with the corresponding values of p_t and s_t , if $p_t + s_t < p_{min} + s_{min}$. The method defines a warning state which is triggered when $p_t + s_t \geq p_{min} + 2 * s_{min}$, and a drift is detected when $p_t + s_t \geq p_{min} + 3 * s_{min}$.

Other methods have modified DDM to enhance its performance for solving diverse tasks. For example, Early Drift Detection Method (EDDM) [94] extends DDM by tracking the distance between two consecutive misclassifications rather than the error rate. This approach was proven to be more efficient than DDM in detecting gradual drifts [95]. Reactive Drift Detection Method (RDDM) [96] mitigates the performance loss problem of DDM, which is due to decreased sensitivity when the concept has a large number of members. RDDM augments DDM by periodically removing old data instances of long concepts. The authors argued that RDDM provides higher or equal global accuracy than DDM and detects drifts earlier in most situations.

Hoeffding Drift Detection Method (HDDM) [97] modifies DDM by using the Hoeffding's inequality [98] to detect substantial changes in the moving average of the performance estimate. The authors proposed two variants of the method, HDDM_A that is suitable to detect sudden drifts, and HDDM_W for gradual drifts. Fast Hoeffding Drift Detection Method (FHDDM) [99] addressed the shortcomings of HDDM caused by high numbers of false positives and false negatives. FHDDM employs a sliding window to compare the maximum overall probability of a correct prediction and the most recent one. Stacking Fast Hoeffding Drift Detection Method (FHDDMS) and Additive FHDDMS (FHDDMS_{add}) [100] extends FHDDM by maintaining windows of different sizes (short and long sliding windows) to detect various types of drift. FHDDMS_{add} uses a binary indicator of classification errors with its summation.

Accurate Concept Drift Detection Method (ACDDM) [101] uses Hoeffding's inequality to analyze the inconsistency of the error rate for detecting concept drift. Lughofer et al. [102] have designed an approach to detect concept drift in semi-supervised and fully unsupervised problems. The authors modified the standard Page-Hinkley test (PHT) [103] to a faded version that outweighs

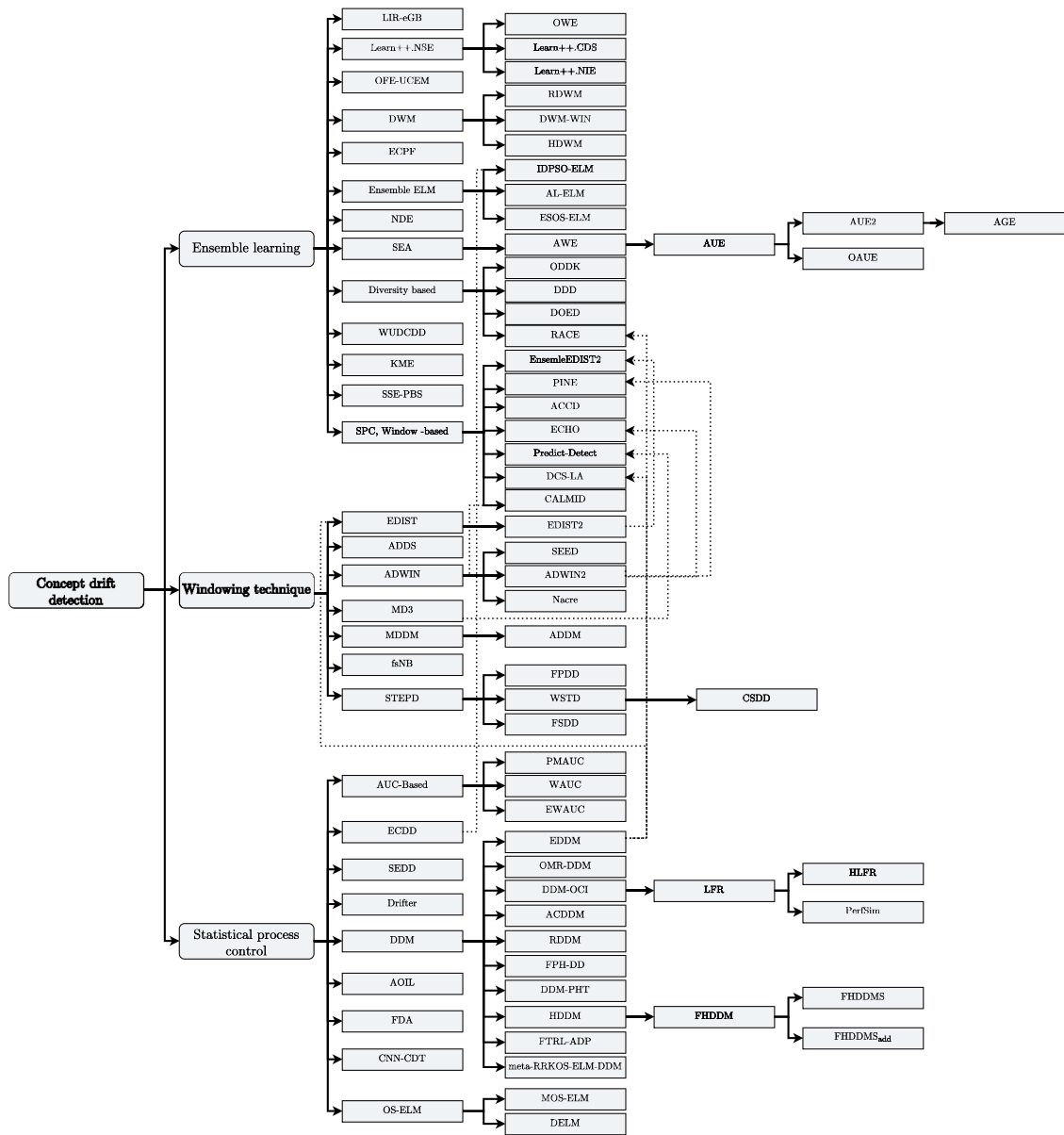


Fig. 9. Classification hierarchy of reviewed performance-based drift detector methods.

older statistics. The PH statistic used to obtain classifier's confidence is based on the Hoeffding bound. Sakamoto et al. [104] have applied DDM to clustering problems by utilizing the assignment error and PHT was used to detect the changes.

DDM was also integrated into more complex frameworks that cope with concept drift. A Meta-cognitive Recurrent Recursive Kernel Online Sequential Extreme Learning Machine with a modified DDM (meta-RRKOS-ELM-DDM) [105] was presented to solve the concept drift problem and reduce the learning time. The authors modified DDM so it could be employed in time series forecasting by calculating the error rate $ER_{l,p}$ and the standard deviation $SD_{l,p}$, for each sample l in step p of the time series prediction. The meta-cognitive learning strategy automatically finds the Approximate Linear Dependence Kernel Filter (ALD) threshold to scale down the computation complexity.

Follow-the-Regularized-Leader with Adaptive Decaying Proximal (FTRL-ADP) [106] is based on Time Decaying Adaptive Prediction (TDAP) algorithm and uses the DDM drift detector to speed up the adaptation to concept drift. This adaptation allows tuning the decaying rate of the TDAP algorithm, automatically. Online

Map-Reduce Drift Detection Method (OMR-DDM) [107] combines the online error rate of parallel classification algorithms to detect drifts using a Map-Reduce framework.

DDM was also modified to be utilized in online class imbalance learning problems. Drift Detection Method for Online Class Imbalance (DDM-OCI) [108] is one of the first algorithms in this category. The method uses the same test statistic as DDM, but tracks the degradation in the minority-class recall to signal concept drift. The method triggers many false alarms in scenarios where the majority-class is affected by the drift since it only considers the true positive rate $P(tp)$. Linear Four Rates (LFR) [109] has improved the limitation of DDM-OCI by monitoring the four rates of the confusion matrix, true positive rate (tp), true negative rate (tn), false positive rate (fp) and false negative rate (fn). Hierarchical Linear Four Rates (HLFR) [110] uses the same four rates as LFR hierarchically in two testing layers. PerfSim [111] handles imbalanced datasets with concept drift by calculating the Cosine Similarity measure of TP and FP of all classes and comparing them to a given threshold α .

Some other testing techniques were applied to monitor the model's performance degradation. Song et al. [112] have proposed fuzzy error deviation (fed) metric, which is computed to estimate the drift severity based on the variation of the predictor error. Adaptive Online Incremental Learning for evolving data streams (AOIL) [113] monitors the change in the mean and variance values of the loss error to detect the drift. Spectral Entropy Drift Detector (SEDD) [114] computes the spectral entropy along the error stream to verify the fluctuation's magnitude along the learning process. The Drifter algorithm [115] calculates the generalization error on the dataset (RMSE) to detect concept drift. The algorithm determines the detection threshold σ using receiver operating characteristics (ROC) analysis.

EWMA for Concept Drift Detection (ECDD) [116] adjusts the conventional exponentially weighted moving average charts (EWMA) [117] to monitor changes in the error rate of the classifier. At time t , the error rate $\hat{p}_{0,t}$, and the dynamic standard deviation σ_{Z_t} of the EWMA estimator Z_T , are calculated. Concept drift is flagged if:

$$Z_t > \hat{p}_{0,t} + L_t \sigma_{Z_t} \quad (6)$$

where the control limit L_t is provided by the authors. Disabato and Roveri [118] have adapted Convolutional Neural Networks (CNN) by incorporating Change Detection Tests (CDTs) based on monitoring the classification error to detect concept drift using CUMulative SUM (CUSUM) test [119]. Other works control different performance metrics to detect drifts. As in [120], authors have proposed a family of AUC-based metrics, namely Prequential Multi-Class AUC (PMAUC), Weighted AUC (WAUC), and Equal Weighted AUC (EWAUC). The metrics can be utilized as a part of the concept drift detection method for multi-class imbalanced data by tracking their values over time.

Extreme learning machine (ELM) [121] was exploited to detect concept drift, in particular, online sequential ELM (OS-ELM) [122]. Yang et al. [123] have proposed a method that can detect concept drift based on the dissimilarities between the output weights of the OS-ELM models for every chunk of new data. Dynamic Extreme Learning Machine (DELM) [124] modifies ELM by adding concept drift detection that monitors the performance degradation of the learner. Based on the result of the detector, DELM will add additional hidden layer nodes in case of concept drift occurrence. Another method that utilizes ELM is the Meta-cognitive online sequential extreme learning machine (MOS-ELM) [125]. MOS-ELM incorporates two tests depending on the type of drift, one for gradual drift and another for sudden drifts. It uses the weighted extreme learning machine (WELM) to track the classification performance in imbalanced datasets.

5.2. Windowing technique

Window-based detectors divide the data stream into windows based on data size or time interval in a sliding manner. These methods monitor the performance of the most recent observations introduced to the learner and compare it with the performance of a reference window.

ADaptive WINdowing (ADWIN) and its extension (ADWIN2) [126] are among the most popular methods that use the windowing technique to detect drifts. ADWIN uses the Hoeffding bound to examine the change between the means, μ_{hist} and μ_{new} , of the two sufficiently large sub-windows, W_{hist} and W_{new} :

$$|\mu_{hist} - \mu_{new}| > 2\epsilon_{cut} \quad (7)$$

where ϵ_{cut} is the optimal cut:

$$\epsilon_{cut} = \sqrt{\frac{1}{2m} \ln \frac{4|W|}{\delta}} \quad (8)$$

where m is the harmonic mean of the two windows, and δ is a pre-defined confidence parameter.

SEED [127] adopts the ADWIN method by comparing two sub-windows within a window W , a left sub-window W_L and right sub-window W_R . SEED monitors a binary sequence of the classification decision, 1 for correct predictions and 0 for errors. The algorithm sets the boundaries of cutting the windows by using the Hoeffding Inequality with Bonferroni correction to calculate ϵ_{cut} , the test statistic to compare the averages of data instances of each window.

Another well-recognized and straightforward method for concept drift detection is STEP D [95], which relies on two-time windows, a recent window r and overall window o . It applies the statistical test of equal proportions to compare the accuracies between the two windows as follows:

$$T(r_o, r_r, n_o, n_r) = \frac{|r_o/n_o - r_r/n_r| - 0.5(1/n_o + 1/n_r)}{\sqrt{\hat{p}(1-\hat{p})(1/n_o + 1/n_r)}} \quad (9)$$

where r is the number of correct predictions, n is the window size, and $\hat{p} = (r_o + r_r) / (n_o + n_r)$. P -value is then calculated and compared with the significance level to signal the drift. Wilcoxon Rank Sum Test Drift Detector (WSTD) [128] was inspired by STEP D and applies Wilcoxon rank sum statistical test [129] to detect the drift and limits the size of the older window. Cabral and Barros [130] have modified STEP D to propose three methods to detect drifts, namely Fisher Proportions Drift Detector (FPDD), Fisher Square Drift Detector (FSDD), and Fisher Test Drift Detector (FTDD). The only difference between these methods and STEP D is that they used Fisher's Exact test [131] to calculate the p -value. Cosine Similarity Drift Detector (CSDD) [132] works similarly to WSTD by calculating the confusion matrix based on the Positive Predictive Value (PPV) and False Discovery (FDR) rates instead of TP and FP for each window, which are calculated as $PPV_r = TP/(TP + FP)$ and $FDR_r = FP/(TP + FP)$. Then the Cosine Similarity is computed between the vectors created from the confusion matrices of the two windows to signal a drift or warning alert. In a recent study [133], authors have proposed the Nacre framework that uses the ADWIN strategy to set the window size in a stability detector that monitors the predictive performance.

Similar practices have been followed to process the window. McDiarmid Drift Detection Method (MDDM) [134] slides a window over the prediction results, 1 for correct predictions, and 0 for false predictions. The entries of the prediction results stream are weighted by recency. The method uses McDiarmid's inequality [135] to determine the significance in the difference between the maximum weighted average seen so far and the weighted mean of entries in the sliding window. ADaptive sliding window-based Detection Method (ADDMM) [136] follows a similar approach as MDDM but monitors the entropy of the prediction results stream over a sliding window.

Other window-based approaches can be found in the literature. The Margin Density Drift Detection (MD3) [137] approach processes the data stream as a sliding window. It monitors the number of samples that fall within the classifier's margins for every chunk of data. The approach triggers a drift alert based on a comparison with the density threshold θ . Fast switch Naive Bayes model (fsNB) [138] performs two-sample Kolmogorov-Smirnov test (KS test) [139] to compare the residuals of a fine-tuned model and a retrained model to decide which model to use. Error DISTance for drift detection and monitoring (EDIST) [140] modifies EDDM by maintaining two data windows, a global sliding window and another one that contains the current examples. EDIST detects the drift by checking if the error distance distributions between the two windows exceed a threshold ϵ . The ϵ value tunes itself adaptively based on the statistical hypothesis test. The experiments showed that the method is robust to noise

and false alarms. Khamassi et al. [141] have extended EDIST by introducing EDIST2, which can handle gradual local drifts by using all the data in relearning the model instead of only using the data window in the drifted region. Anti-concept Drift Detection Algorithm (ADDS) [142] applies Hoeffding's inequality to track the difference between the optimal accuracy and the real-time accuracy in a sliding window. ADDS concludes that the error in the classification accuracy should be within a threshold $\epsilon = \sqrt{\frac{1}{2n} \ln \frac{1}{\delta}}$, where n is the sliding window size and δ is the confidence level, otherwise concept drift is detected.

5.3. Ensemble learning

Concept drift detectors that are ensemble-based operate by combining the results of multiple diverse base learners. The overall performance is monitored by either considering the accuracy of all the ensemble members or the accuracy of each individual base learner. Note that this is different from an ensemble of drift detectors as in [143,144], where the decisions of multiple drift detectors are combined to signal the drift. Experimental studies demonstrate that an ensemble of drift detectors does not guarantee higher performance than the individual detection methods [145].

Ensemble-based detectors trigger concept drift if the learners suffer from a significant level of performance degradation. This assumption is based on the fact that each learner has capabilities in solving specific problems [44]. Most of the ensemble-based detectors are built upon the Weighted Majority Algorithm (WMA) [146] method. WMA elects the best learners in the ensemble by giving each one a weight based on its performance. Streaming Ensemble Algorithm (SEA) [147] approach is one of the earliest ensemble-based works to tackle concept drift. SEA handles the drift implicitly by creating a new learner for each new chunk of the data till the maximum number of learners is reached. The learners are refined based on their prediction performance. A similar method in refining the ensemble was proposed in the Accuracy Weighted Ensemble (AWE) [148]. The novelty of AWE is in selecting the best learners by using a special version of the mean squared error that deals with probabilities to select the best n learners and discard outdated learners with the highest performance degradation rate. Brzezinski and Stefanowski [149] have proposed Accuracy Updated Ensemble (AUE) algorithm, which improves AWE by conditionally updating the component learners rather than only regulating the weights. The authors also used a simpler weighting function than the one in AWE. AUE2 [88] improved AUE by introducing a cost-effective weight and pruning base learners. Online Accuracy Updated Ensemble (OAUE) [150] utilizes a drift detector included in an online learner that triggers a reweighting signal to the learner. Accuracy and Growth Rate updated Ensemble (AGE) [151] has extended AUE2 to react to various types of drift. AGE uses the geometric mean to design the Growth Rate of base learners.

Dynamic Weighted Majority (DWM) [61] is one of the most popular passive ensemble approaches, which employs a weighting mechanism inspired by WMA. Every learner's weight is reduced by a multiplicative factor β , $0 < \beta \leq 1$, when it gives a wrong prediction every ρ time step. To overcome the drawback of DWM, which does not consider the learner's performance on the training data, DWM-WIN was proposed in [152]. DWM-WIN is an ensemble method that includes the learner's age in the weighting mechanism and tracks the concept drift in the learning phase. In recent research, Heterogeneous Dynamic Weighted Majority (HDWM) [153] was proposed to turn DWM into a heterogeneous ensemble by automatically choosing the best learners to be used over time to prevent performance degradation. Recurring Dynamic Weighted Majority (RDWM) [154] is built upon DWM

by forming two ensembles of learners. The primary ensemble represents the current concepts, and the secondary ensemble consists of the most accurate learners.

Another well-known ensemble-based drift detection method is Learn++.NSE (incremental learning for NSEs) [20]. Learn++.NSE is the first version of the notable set of ensemble algorithms Learn++ [155] to address concept drift. In Learn++.NSE, a set of learners is trained on chunks of data examples. The training examples are weighted according to the ensemble error on this example. If the example is correctly classified by the ensemble i , Learn++.NSE sets its weight to 1, otherwise it is penalized to $w_i = 1/e$. The sigmoid function is used to weigh the learners in the ensemble based on their errors on the old and current chunks. Ditzler and Polikar [156] have proposed a framework that includes two related ensemble-based approaches, namely Learn++.CDS and Learn++.NIE. They extended their prior work on Learn++.NSE to accommodate class-imbalanced data. The methods monitor the performance of both the majority and minority classes. On-line Weighted Ensemble (OWE) [157] was proposed to adapt Learn++ for regression tasks.

Other methods use the diversity between the learners in the ensemble. Diversity for Dealing with Drifts (DDD) [158] controls the diversity level of the learners in the ensemble by incorporating both low diversity and high diversity ensembles. The low diversity ensemble is used to detect the drift, and the high diversity ensemble is used after detecting the drift. Diversified Online Ensembles Detection (DOED) [159] develops two ensembles with different levels of diversity, **E0** and **E1**. DOED uses only one significance level to detect concept drift with **E0** and **E1** using the P -value. If any of the ensembles detect a drift, the ensemble is re-initialized. If both detect the drift, the ensemble with the lower accuracy is re-initialized. Recurrent Adaptive Classifier Ensemble (RACE) [160] preserves an archive of diverse learners and uses EDDM to detect recurring drifts. The online drift detector for the K -class problem (ODDK) [161] was proposed to handle multi-class problems with concept drift. The algorithm constructs a contingency table that stores the variation of the diversity of a pair of classifiers and uses the PH test to detect concept drift.

The benchmark methods of the other categories, statistical process control and windowing technique, were also used in ensemble frameworks. Pinag  et al. [162] have modified DDM and EDDM to work as unsupervised detection methods producing a *pseudo prequential error rate* that is monitored for every ensemble member by assuming the predicted value is the true label. The drift is detected if n members of the ensemble reach a drift level. Predictive and parameter INsensitive Ensemble (PINE) [163] is an ensemble approach that processes asynchronous concept drifts in classification in distributed networks. A modified version of the ADWIN drift detector is provided for each peer of the framework. The detector monitors a stream of accuracies represented by ones and zeros. More recently, Liuet al. [164] have proposed CALMID method for multiclass imbalanced streaming data with concept drift that uses ADWIN algorithm in ensemble settings. Associative Classification over Concept Drifting Data Streams (ACCD) [165] checks the current accuracy of an ensemble of online classifiers by comparing it with the estimated statistical lower bound of maximum accuracy to signal a drift. EnsembleEDIST2 [166] makes use of EDIST2 as a drift detector in the proposed ensemble-based drift handling approach to track the learners' performance.

Predict-Detect streaming framework [167] relies on detecting adversarial drifts from unlabeled data streams inspired by the MD3 framework. The framework uses the training data to learn the expected disagreement PD_{Ref} and accepted deviation σ_{Ref} of the ensemble. An adversarial drift is detected if a sudden increase in the disagreement metric PD occurs. Efficient Concept Drift and Concept Evolution Handling over Stream Data (ECHO) [168]

is a semi-supervised ensemble-based framework that contains a concept drift detection technique. ECHO maintains a sliding window over the data stream to monitor significant changes in the classifier's confidence to detect concept drift using the CUSUM test. Khezri et al. [169] proposed an ensemble-based Performance-Based Selection (PBS) metric for semi-supervised learning problems with concept drift. The model performance is evaluated based on pseudo-accuracy and energy regularization.

ELM has also been employed in the ensemble approach to deal with concept drift. An ensemble of online sequential extreme learning machines (ESOS-ELM) [170] was proposed to tackle concept drift in class imbalance data. ESOS-ELM maintains an ensemble of OS-ELMs and monitors the error rate using a threshold-based technique. In [171], authors have developed two approaches IDPSO-ELM-B and IDPSO-ELM-S to detect concept drift in time series forecasting. The approaches were built upon the swarm behavior of ELM by using the ECDD approach. Xu et al. [172] have proposed an alternating learners framework that uses a drift detector and employs ELM as a base learner for regression problems.

Other strategies were proposed in an ensemble learning framework to deal with concept drift. Number and Distance of Errors (NDE) [173] is an ensemble method that detects concept drift based on the number and distance between the errors and compares it with a threshold. Knowledge-maximized ensemble (KME) [174] is a concept-drift-detection system that contains a $TEST_i$ concept drift detector which checks if the classification error of the ensemble falls below the confidence interval in a sliding window. Enhanced Concept Profiling Framework (ECPF) [175] is a meta-learning framework that tracks the learner behavior to detect changes. Wang et al. have proposed a new pruning criterion, called the loss improvement ratio (LIR), for performance evaluation that is utilized in a pruning strategy to remove outdated learners. The meta-learner decides if the current learner should be reused or replaced based on the performance. Weighted classification and Update algorithm of Data stream based on Concept Drift Detection (WUDCDD) [176] approach signals a drift warning if the performance degrades for the current data chunks. The system signals a drift detection alert if the degradation is still present. The method calculates the Mahalanobis distance between the classification error rate on the data blocks. [93] uses the Uncertainty Error Correlation Matrix (UECM) to detect concept drift and give each online learner a corresponding weight. UECM is constructed from the error value of the online learning algorithms in the ensemble, and each entry of the matrix represents the strength between the loss function of each learner.

6. Analysis and discussion

This section analyzes and discusses the relevant works we have reviewed to draw conclusions and accentuate the trends in this research area. The analysis is based on spotlighting the main facets that characterize the methods presented in this paper. To facilitate answering **RQ3**, we have steered our attention to investigate multiple attributes that were handled in designing the methods. The attributes are: (1) the machine learning problem managed by the method, (2) the performance metric used to track model degradation, (3) the base learner employed in the predictive system, and (4) the type of drift addressed by the approach.

6.1. Machine learning problem scope

Narrowing down the scope of the machine learning problem is a fundamental step in designing the concept drift detection method since each learning problem requires calculating different

performance metrics. In Fig. 10 we summarized the machine learning scope of the surveyed methods. We can see that drift detection for classification tasks is the main scope in the literature, while few approaches address the regression settings. The main reason for that is the lack of relevant datasets for regression problems with concept drift [177], and the wide availability of classification datasets for concept drift detection purposes [18]. There is also a reasonable number of studies in the area of class imbalance problems since the concept drift, and class imbalance problems are closely related and affect each other [43]. More recently, performance-based drift detection methods have been proposed in the context of more complicated, semi-supervised, and unsupervised problems. These problems pose a significant challenge for performance-based detectors since the ground truth labels are not provided. Semi-supervised detectors usually operate by predicting the labels of the unlabeled examples and proceed with computing the performance loss to detect drift [178]. On the contrary, the unsupervised drift detector approach tries to estimate a pseudo-error and self-evaluate the performance [179].

Most of the proposed approaches are devoted to solving a specific problem. This confirms that concept drift detection adheres to the No Free Lunch Theorem [180], and a universal approach that copes with all machine learning problems are challenging to find [38].

6.2. Performance metrics

As shown in Fig. 11, the majority of the methods rely on the classification error rate to detect the degradation in the predictive performance. This could be because most approaches have been evaluated within the classification task context, which received most of the attention in the literature. In addition, calculation of classification error rate metric entails low complexity and cost needed. Since the accuracy is not always indicative of performance loss, drift detectors are criticized for a high number of false alarms. For class imbalance tasks, and since accuracy is not an expressive metric for performance, authors have adopted other metrics such as confusion matrix and AUC. Some studies have designed drift detectors based on metrics calculated from the model's intrinsic behavior, such as performance gain and growth rate. We have grouped these model-wise metrics into a model-based category.

6.3. Base learners

Drift detection systems require many updates once they are deployed. Consequently, drift detection methods should support incremental learning and adapt dynamically. For that reason, Hoeffding Trees (HT) and Naive Bayes (NB) are adopted as base learners for the majority of performance-based concept drift detectors. Furthermore, HT and NB also have sufficient capability to learn from and deal with massive data streams. More recent works have adopted neural networks within their framework. Still, these approaches could make deploying within big data stream systems challenging since it is difficult to update the neural network architecture dynamically. Another major drawback for neural networks is the lack of transparency and interpretability [181,182]. This drawback causes a burden to concept drift handling systems since drift understanding plays a significant part in detecting and adapting to drifts [18,183]. This Fig. 12 summarizes the base learners used in the reviewed approaches. We use *model-agnostic* only for those papers which explicitly stated that the detection method could be integrated with any predictive model. Otherwise, we use the base learner as reported in the original study.

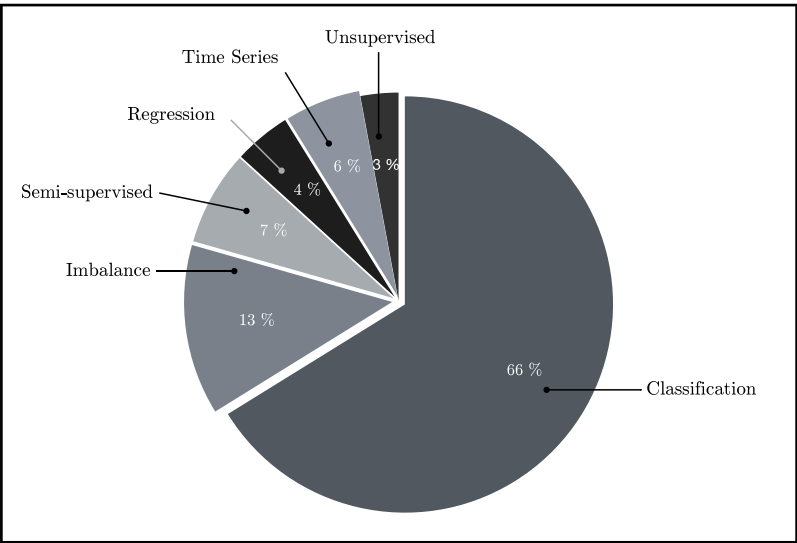


Fig. 10. Machine learning problem scope of the methods.

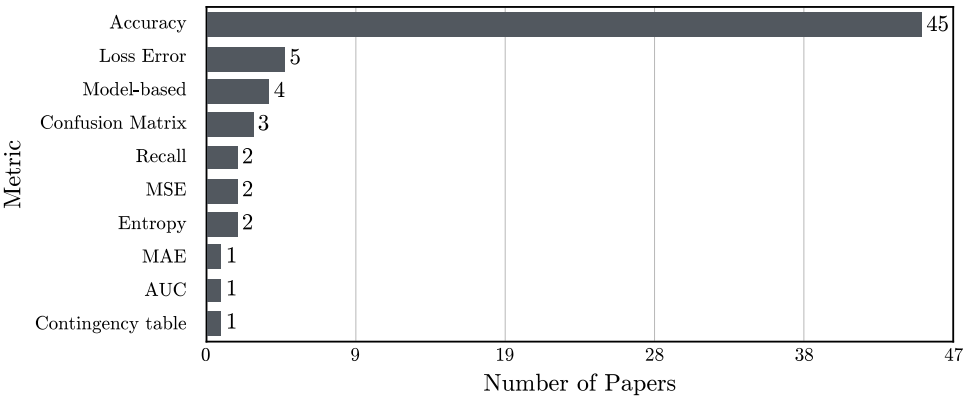


Fig. 11. Performance metrics monitored in the methods.

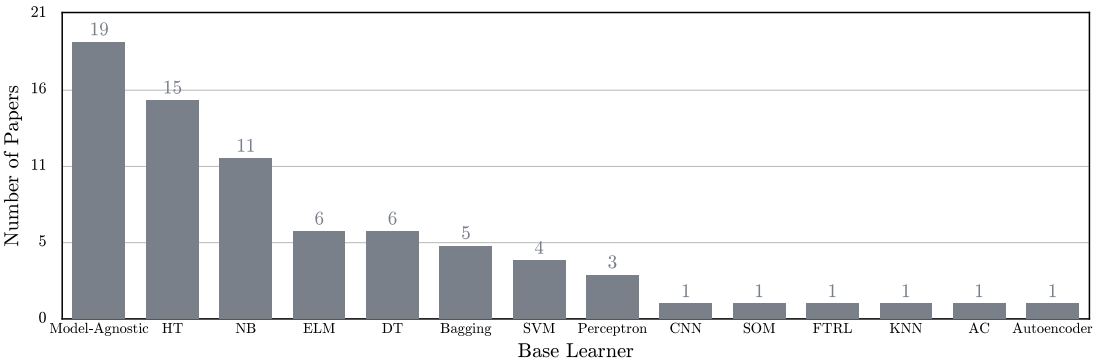


Fig. 12. Base learners adopted in the methods.

6.4. Drift types

Table 3 provides an overview of the methods that explicitly mentioned the handled drift type. Sudden and gradual drifts are the main drift types addressed in the reviewed literature. While fewer works addressed the incremental drift, it is not always easy to distinguish between the natural evolution of systems and continuous changes. Recurring drift must be processed in a specific way, where the system must be supplied with a buffer to

store the old behavior and reuse the learned knowledge from the past observations once it reappears in the system.

7. Conclusions and future directions

Concept drift and performance degradation are two intertwined phenomena in predictive systems. The existence of one phenomenon articulates the other. In this paper, we presented a comprehensive and up-to-date overview of the concept drift

Table 3

Summary of the methods with the handled drift type. The method names in italics were proposed by authors since there was no name given in the original paper.

Method	Year	Drift type			
		Sudden	Gradual	Incremental	Recurring
SSE-PBS [169]	2021	✓	✓		
ODKK [161]	2021	✓	✓		✓
RACE [160]	2021				✓
LIR-eGB [184]	2021	✓	✓	✓	
CALMID [164]	2021	✓	✓	✓	✓
Nacre [133]	2021				✓
SEDD [114]	2021	✓	✓		
OFE-UECM [93]	2020	✓	✓	✓	✓
FDA [112]	2020	✓		✓	
ACDDM [101]	2020	✓	✓	✓	✓
HDWM [153]	2020	✓	✓		✓
OS-ELMs [123]	2020	✓		✓	
DCS-LA [162]	2020	✓	✓		
HLFR [110]	2019	✓	✓	✓	✓
RDWM [154]	2019	✓	✓	✓	✓
CSDD [132]	2019	✓	✓		
ECPF [175]	2019				✓
FHDDMS, FHDDMS _{add} [100]	2018	✓	✓		
FPDD, FSDD [130]	2018	✓	✓		
FTRL-ADP [106]	2018	✓	✓	✓	✓
KME-TEST _i [174]	2018	✓	✓	✓	✓
WSTD [128]	2018	✓	✓		
MDDM [134]	2018	✓	✓		
RDDM [96]	2017	✓	✓		
ADDS [142]	2017	✓	✓		
AL-ELM [172]	2017	✓	✓		
FPH-DD [102]	2016	✓	✓		
MOS-ELM [125]	2016	✓	✓		
NDE [173]	2016	✓	✓		
FHDDM [99]	2016	✓	✓		
DOED [159]	2015	✓	✓	✓	✓
HDDM [97]	2015	✓	✓		
ESOS-ELM [170]	2015	✓	✓		
LFR [109]	2015	✓	✓		
DDM-PHT [104]	2015	✓	✓	✓	
EDIST [140]	2014	✓	✓		
OAE [150]	2014	✓	✓	✓	✓
SEED [127]	2014	✓	✓		
AGE [151]	2014	✓	✓		
ACCD [165]	2014	✓	✓		
ADDM [136]	2014	✓	✓		
AUE2 [150]	2014	✓	✓		
LEARN++-CDS [156]	2013	✓		✓	✓
ECDD [116]	2012	✓	✓		

research field. We started by describing the main causes of concept drift, followed by common definitions and measures of concept drift. We have compiled the various terms used in the literature to refer to concept drift types since the area is deluged with terminologies. We then presented concept drift detection approaches that track the performance degradation to identify changes. These performance-based methods work reversely by signaling concept drift when the performance degrades to a certain threshold. Real concept drift leads to deterioration in the predictive accuracy as it requires adaptation to changes. The findings of this study are extracted and summarized in the following points:

1. Multiple terms can be found in the literature for the same concept drift type. Also, the same term is used for multiple concept drift types. Therefore we suggest using the mathematical definition to refer to specific concept drift types.
2. The classification problem comprises the major part of the task scope in drift handling. A limited number of works have been developed to undertake other scopes.

3. Most existing performance-based detectors rely on monitoring the error rate to identify the performance degradation and trigger a drift; recent advances have monitored new performance metrics.
4. Performance-based detection methods have been used in unsupervised and semi-supervised learning by introducing new metrics to evaluate the model's performance, such as *pseudo-error*.
5. Most of the designed solutions used Hoeffding Trees or Naive Bayes algorithm as base learners. While employing neural networks has recently started to emerge.
6. There is still no clear evidence about the ideal drift detector to be used in a specific problem or setting.

Based on the mentioned findings, we suggest the following future research directions:

1. Since few methods deal with regression settings, more research on detecting drifts in regression scope is highly desired. It is considered one of the main tasks in machine learning and is now employed in a wide range of applications [185].

2. As previously proven in the comparison studies [5,186], there is no single drift detector that works better than all the others in all scenarios. It would be interesting to evaluate the methods against different datasets and investigate their applicability in specific domains. This would support users in selecting the suitable method for the problem at hand.
3. Most of the existing methods in the literature suffer from a high number of false alarms. This is because most of the approaches are over-reliant on monitoring the degradation in the learner's accuracy. A multiple hypothesis technique could be a solution by monitoring other metrics to have a stronger assumption on drift detection.
4. The extensive research conducted on incremental and on-line learning paradigms could be leveraged in drift detection methods by employing the recent advances in drift handling systems [187]. Since these paradigms are characterized by high capabilities in continuously adapting to accommodate the incoming data points [188].
5. Another opportunity would be utilizing the staggering progress in the explainable deep learning field that has been recently achieved [189,190]. These explainable models would make efficient deep learning more useful in understanding and handling concept drift.

CRedit authorship contribution statement

Firas Bayram: Conceptualization, Methodology, Writing – original draft, Visualization, Investigation. **Bestoun S. Ahmed:** Supervision, Writing – original draft, Investigation, Writing – review & editing. **Andreas Kassler:** Supervision, Writing – review & editing.

Declaration of competing interest

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

Acknowledgment

Parts of this work has been funded by the Knowledge Foundation of Sweden (KKS) through the Synergy Project AIDA – A Holistic AI-driven Networking and Processing Framework for Industrial IoT (Rek:20200067).

References

- [1] G. Marcus, Deep learning: A critical appraisal, 2018, arXiv preprint arXiv:1801.00631.
- [2] G.M. Weiss, Mining with rarity: A unifying framework, *ACM SIGKDD Explor. Newsl.* 6 (1) (2004) 7–19.
- [3] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, *Mach. Learn.* 23 (1) (1996) 69–101.
- [4] T.R. Hoens, R. Polikar, N. Chawla, Learning from streaming data with concept drift and imbalance: an overview, *Prog. Artif. Intell.* 1 (2011) 89–101.
- [5] R.S.M. de Barros, S.G.T. de Carvalho Santos, An overview and comprehensive comparison of ensembles for concept drift, *Inf. Fusion* 52 (2019) 213–244.
- [6] M. Asghari, D. Sierra-Sosa, M. Telahun, A. Kumar, A.S. Elmaghraby, Aggregate density-based concept drift identification for dynamic sensor data models, *Neural Comput. Appl.* 33 (8) (2021) 3267–3279.
- [7] R. Xu, Y. Cheng, Z. Liu, Y. Xie, Y. Yang, Improved long short-term memory based anomaly detection with concept drift adaptive method for supporting IoT services, *Future Gener. Comput. Syst.* 112 (2020) 228–242.
- [8] G. Fenza, M. Gallo, V. Loia, Drift-aware methodology for anomaly detection in smart grid, *IEEE Access* 7 (2019) 9645–9657.
- [9] M. Mohammadpourfard, Y. Weng, M. Pechenizkiy, M. Tajdianian, B. Mohammadi-Ivatloo, Ensuring cybersecurity of smart grid against data integrity attacks under concept drift, *Int. J. Electr. Power Energy Syst.* 119 (2020) 105947.
- [10] S.K. Perepu, K. Dey, CDDM: A method to detect and handle concept drift in dynamic mobility model for seamless 5G services, in: 2020 IEEE Globecom Workshops (GC Wkshps), 2020, pp. 1–6.
- [11] Y. Hu, K. Liu, X. Zhang, K. Xie, W. Chen, Y. Zeng, M. Liu, Concept drift mining of portfolio selection factors in stock market, *Electron. Commer. Res. Appl.* 14 (6) (2015) 444–455.
- [12] A.L. Suárez-Cetrulo, A. Cervantes, D. Quintana, Incremental market behavior classification in presence of recurring concepts, *Entropy* 21 (1) (2019).
- [13] Y. Xu, K. Wilson, Early alert systems during a pandemic: A simulation study on the impact of concept drift, in: LAK21: 11th International Learning Analytics and Knowledge Conference, Association for Computing Machinery, New York, NY, USA, 2021, pp. 504–510.
- [14] I. Zliobaite, M. Pechenizkiy, J. Gama, An Overview of Concept Drift Applications, in: *Big Data Analysis: New Algorithms for a New Society*, vol. 16, Springer International Publishing, 2016, pp. 91–114.
- [15] G. Ditzler, M. Roveri, C. Alippi, R. Polikar, Learning in nonstationary environments: A survey, *IEEE Comput. Intell. Mag.* 10 (4) (2015) 12–25.
- [16] M. Basseville, I.V. Nikiforov, Detection of Abrupt Changes: Theory and Application, Prentice-Hall, Inc., USA, 1993.
- [17] R. Pears, S. Sakthithasan, Y.S. Koh, Detecting concept change in dynamic data streams, *Mach. Learn.* 97 (3) (2014) 259–293.
- [18] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under concept drift: A review, *IEEE Trans. Knowl. Data Eng.* 31 (12) (2019) 2346–2363.
- [19] S. Grossberg, Nonlinear neural networks: Principles, mechanisms, and architectures, *Neural Netw.* 1 (1) (1988) 17–61.
- [20] R. Elwell, R. Polikar, Incremental learning of concept drift in nonstationary environments, *IEEE Trans. Neural Netw.* 22 (10) (2011) 1517–1531.
- [21] J.L. Lobo, J. Del Ser, F. Herrera, LUNAR: Cellular automata for drifting data streams, *Inform. Sci.* 543 (2021) 467–487.
- [22] Y. Song, J. Lu, H. Lu, G. Zhang, Learning data streams with changing distributions and temporal dependency, *IEEE Trans. Neural Netw. Learn. Syst.* (2021).
- [23] A. Dries, U. Rückert, Adaptive concept drift detection, *Stat. Anal. Data Min.* 2 (5–6) (2009) 311–327.
- [24] D. Kifer, S. Ben-David, J. Gehrke, Detecting change in data streams, in: *Proceedings of the Thirtieth International Conference on Very Large Data Bases – Volume 30, VLDB Endowment*, 2004, pp. 180–191.
- [25] I. Goldenberg, G.I. Webb, Survey of distance measures for quantifying concept drift and shift in numeric data, *Knowl. Inf. Syst.* (2018) 1–25.
- [26] J. Gama, I. Žliobaitundefined, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, *ACM Comput. Surv.* 46 (4) (2014).
- [27] J. Gama, R. Sebastiao, P.P. Rodrigues, On evaluating stream learning algorithms, *Mach. Learn.* 90 (3) (2013) 317–346.
- [28] R. Sebastiao, J. Gama, A study on change detection methods, in: *Progress in Artificial Intelligence*, 14th Portuguese Conference on Artificial Intelligence, EPIA, 2009, pp. 12–15.
- [29] T.M. Mitchell, Machine Learning, first ed., 1997.
- [30] J. Gama, P. Medas, G. Castillo, P. Rodrigues, Learning with drift detection, in: *Advances in Artificial Intelligence – SBIA 2004*, 2004, pp. 286–295.
- [31] N. Lu, G. Zhang, J. Lu, Concept drift detection via competence models, *Artificial Intelligence* 209 (2014) 11–28.
- [32] J. Demšar, Z. Bosnić, Detecting concept drift in data streams using model explanation, *Expert Syst. Appl.* 92 (2018) 546–559.
- [33] J.L. Lobo, J. Del Ser, I. Laña, M.N. Bilbao, N. Kasabov, Drift detection over non-stationary data streams using evolving spiking neural networks, in: *Intelligent Distributed Computing XII*, Springer International Publishing, 2018, pp. 82–94.
- [34] D.T.J. Huang, Y.S. Koh, G. Dobbie, A. Bifet, Drift detection using stream volatility, in: A. Appice, P.P. Rodrigues, V. Santos Costa, C. Soares, J. Gama, A. Jorge (Eds.), *Machine Learning and Knowledge Discovery in Databases*, Springer International Publishing, Cham, 2015, pp. 417–432.
- [35] A. Seeliger, T. Nolle, M. Mühlhäuser, Detecting concept drift in processes using graph metrics on process graphs, in: *Proceedings of the 9th Conference on Subject-Oriented Business Process Management*, Association for Computing Machinery, New York, NY, USA, 2017.
- [36] R. Paudel, W. Eberle, An approach for concept drift detection in a graph stream using discriminative subgraphs, *ACM Trans. Knowl. Discov. Data* 14 (6) (2020).
- [37] D. Zambon, C. Alippi, L. Livi, Concept drift and anomaly detection in graph streams, *IEEE Trans. Neural Netw. Learn. Syst.* 29 (11) (2018) 5592–5605.
- [38] H. Hu, M. Kantardzic, T.S. Sethi, No free lunch theorem for concept drift detection in streaming data classification: A review, *Wiley Interdiscip. Rev.: Data Min. Knowl. Discov.* 10 (2) (2020) e1327.
- [39] I. Khamassi, M.S. Mouchaweh, M. Hammami, K. Ghédira, Discussion and review on evolving data streams and concept drift adapting, *Evol. Syst.* 9 (2018) 1–23.
- [40] S. Wares, J. Isaacs, E. Elyan, Data stream mining: methods and challenges for handling concept drift, *SN Appl. Sci.* 1 (2019).
- [41] A.S. Iwashita, J.P. Papa, An overview on concept drift learning, *IEEE Access* 7 (2019) 1532–1547.

- [42] R.N. Gemaque, A.F.J. Costa, R. Giusti, E. Santos, An overview of unsupervised drift detection methods, *Wiley Interdiscip. Rev.: Data Min. Knowl. Discov.* 10 (2020).
- [43] S. Wang, L.L. Minku, X. Yao, A systematic study of online class imbalance learning with concept drift, *IEEE Trans. Neural Netw. Learn. Syst.* 29 (10) (2018) 4802–4821.
- [44] B. Krawczyk, L.L. Minku, J. Gama, J. Stefanowski, M. Woźniak, Ensemble learning for data stream analysis: A survey, *Inf. Fusion* 37 (2017) 132–156.
- [45] J.G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N.V. Chawla, F. Herrera, A unifying view on dataset shift in classification, *Pattern Recognit.* 45 (1) (2012) 521–530.
- [46] J. Gama, *Knowledge Discovery from Data Streams*, first ed., Chapman & Hall/CRC, 2010.
- [47] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence, *Dataset Shift in Machine Learning*, The MIT Press, 2009.
- [48] G.I. Webb, R. Hyde, H. Cao, H.L. Nguyen, F. Petitjean, Characterizing concept drift, *Data Min. Knowl. Discov.* 30 (4) (2016) 964–994.
- [49] R. Klinkenberg, Learning drifting concepts: Example selection vs. example weighting, *Intell. Data Anal.* 8 (3) (2004) 281–300.
- [50] Y. Song, J. Lu, A. Liu, H. Lu, G. Zhang, A segment-based drift adaptation method for data streams, *IEEE Trans. Neural Netw. Learn. Syst.* (2021).
- [51] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification* (2nd Edition), Wiley-Interscience, USA, 2000.
- [52] G. Forman, Tackling concept drift by temporal inductive transfer, in: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, in: SIGIR '06, Association for Computing Machinery, New York, NY, USA, 2006, pp. 252–259.
- [53] L.L. Minku, A.P. White, X. Yao, The impact of diversity on online ensemble learning in the presence of concept drift, *IEEE Trans. Knowl. Data Eng.* 22 (5) (2010) 730–742.
- [54] H. Shimodaira, Improving predictive inference under covariate shift by weighting the log-likelihood function, *J. Statist. Plann. Inference* 90 (2) (2000) 227–244.
- [55] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, F. Herrera, A survey on data preprocessing for data stream mining: Current status and future directions, *Neurocomputing* 239 (2017) 39–57.
- [56] S.J. Delany, P. Cunningham, A. Tsybal, L. Coyle, A case-based technique for tracking concept drift in spam filtering, *Knowl.-Based Syst.* 18 (4) (2005) 187–195.
- [57] A. Tsybal, M. Pechenizkiy, P. Cunningham, S. Puuronen, Dynamic integration of classifiers for handling concept drift, *Inf. Fusion* 9 (1) (2008) 56–68, Special Issue on Applications of Ensemble Methods.
- [58] M.M. Masud, Q. Chen, J. Gao, L. Khan, J. Han, B. Thuraisingham, Classification and novel class detection of data streams in a dynamic feature space, in: J.L. Balcázar, F. Bonchi, A. Gionis, M. Sebag (Eds.), *Machine Learning and Knowledge Discovery in Databases*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 337–352.
- [59] P. Zhang, X. Zhu, Y. Shi, Categorizing and mining concept drifting data streams, in: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in: KDD '08, Association for Computing Machinery, New York, NY, USA, 2008, pp. 812–820.
- [60] G.I. Webb, L.K. Lee, B. Goethals, F. Petitjean, Analyzing concept drift and shift from sample data, *Data Min. Knowl. Discov.* 32 (5) (2018) 1179–1199.
- [61] J.Z. Kolter, M.A. Maloof, Dynamic weighted majority: An ensemble method for drifting concepts, *J. Mach. Learn. Res.* 8 (2007) 2755–2790.
- [62] N.A. Syed, H. Liu, K.K. Sung, Handling concept drifts in incremental learning with support vector machines, in: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in: KDD '99, Association for Computing Machinery, New York, NY, USA, 1999, pp. 317–321.
- [63] G. Widmer, M. Kubat, Effective learning in dynamic environments by explicit context tracking, in: P.B. Brazdil (Ed.), *Machine Learning: ECML-93*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1993, pp. 227–243.
- [64] A. Tsybal, *The Problem of Concept Drift: Definitions and Related Work*, Computer Science Department, Trinity College Dublin, 2004.
- [65] F. Fdez-Riverola, E. Iglesias, F. Díaz, J. Méndez, J. Corchado, Applying lazy learning algorithms to tackle concept drift in spam filtering, *Expert Syst. Appl.* 33 (1) (2007) 36–48.
- [66] M. Sugiyama, M. Kawanabe, *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*, The MIT Press, 2012.
- [67] G. Kreml, V. Hofer, Classification in presence of drift and latency, in: *2011 IEEE 11th International Conference on Data Mining Workshops*, 2011, pp. 596–603.
- [68] M.M. Lazarescu, S. Venkatesh, H.H. Bui, Using multiple windows to track concept drift, *Intell. Data Anal.* 8 (1) (2004) 29–59.
- [69] J. Gao, W. Fan, J. Han, P.S. Yu, A general framework for mining concept-drifting data streams with skewed distributions, in: *Proceedings of the 2007 Siam International Conference on Data Mining*, SIAM, 2007, pp. 3–14.
- [70] M. Salganicoff, Tolerating concept and sampling shift in lazy learning using prediction error context switching, *Artif. Intell. Rev.* 11 (1–5) (1997) 133–155.
- [71] V. Hofer, G. Kreml, Drift mining in data: A framework for addressing drift in classification, *Comput. Statist. Data Anal.* 57 (1) (2013) 377–391.
- [72] Z. Lipton, Y.-X. Wang, A. Smola, Detecting and correcting for label shift with black box predictors, in: *International Conference on Machine Learning*, PMLR, 2018, pp. 3122–3130.
- [73] K. Azizzadenesheli, A. Liu, F. Yang, A. Anandkumar, Regularized learning for domain adaptation under label shifts, 2019, *ArXiv abs/1903.09734*.
- [74] K. Zhang, B. Schölkopf, K. Muandet, Z. Wang, Domain adaptation under target and conditional shift, in: S. Dasgupta, D. McAllester (Eds.), *Proceedings of the 30th International Conference on Machine Learning*, in: *Proceedings of Machine Learning Research*, vol. 28, PMLR, Atlanta, Georgia, USA, 2013, pp. 819–827.
- [75] A. Subbaswamy, P. Schulam, S. Saria, Preventing failures due to dataset shift: Learning predictive models that transport, in: *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019, pp. 3118–3127.
- [76] T.J.T. Heiser, M.-L. Allikivi, M. Kull, Shift happens: Adjusting classifiers, in: U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, C. Robardet (Eds.), *Machine Learning and Knowledge Discovery in Databases*, Springer International Publishing, Cham, 2020, pp. 55–70.
- [77] T.S. Sethi, M. Kantardzic, H. Hu, A grid density based framework for classifying streaming data in the presence of concept drift, *J. Intell. Inf. Syst.* 46 (1) (2016) 179–211.
- [78] T.D. Nguyen, M. Christoffel, M. Sugiyama, Continuous target shift adaptation in supervised learning, in: G. Holmes, T.-Y. Liu (Eds.), *Asian Conference on Machine Learning*, in: *Proceedings of Machine Learning Research*, vol. 45, PMLR, Hong Kong, 2016, pp. 285–300.
- [79] P. Vorburger, A. Bernstein, Entropy-based concept shift detection, in: *Sixth International Conference on Data Mining (ICDM'06)*, 2006, pp. 1113–1118.
- [80] M.G. Kelly, D.J. Hand, N.M. Adams, The impact of changing populations on classifier performance, in: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in: KDD '99, Association for Computing Machinery, New York, NY, USA, 1999, pp. 367–371.
- [81] N. Charoenphakdee, M. Sugiyama, Positive-unlabeled classification under class prior shift and asymmetric error, in: *Proceedings of the 2019 SIAM International Conference on Data Mining*, SIAM, 2019, pp. 271–279.
- [82] Y. Zhong, H. Yang, Y. Zhang, P. Li, C. Ren, Long short-term memory self-adapting online random forests for evolving data stream regression, *Neurocomputing* 457 (2021) 265–276.
- [83] K.O. Stanley, *Learning Concept Drift with a Committee of Decision Trees*, Informe Técnico: UT-AI-TR-03-302, Department of Computer Sciences, University of Texas at Austin, USA, 2003.
- [84] R.J. Hickey, M.M. Black, Refined time stamps for concept drift detection during mining for classification rules, in: J.F. Roddick, K. Hornsby (Eds.), *Temporal, Spatial, and Spatio-Temporal Data Mining*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 20–30.
- [85] P.M. Gonçalves Jr., R.S.M. de Barros, RCD: A recurring concept drift framework, *Pattern Recognit. Lett.* 34 (9) (2013) 1018–1025.
- [86] R.P.J.C. Bose, W.M.P. van der Aalst, I. Žliobaitė, M. Pechenizkiy, Handling concept drift in process mining, in: H. Mouratidis, C. Rolland (Eds.), *Advanced Information Systems Engineering*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 391–405.
- [87] I. Žliobaitė, *Learning under concept drift: an overview*, 2010, *arXiv preprint arXiv:1010.4784*.
- [88] D. Brzezinski, J. Stefanowski, Reacting to different types of concept drift: The accuracy updated ensemble algorithm, *IEEE Trans. Neural Netw. Learn. Syst.* 25 (1) (2013) 81–94.
- [89] M. Black, R.J. Hickey, Maintaining the performance of a learned classifier under concept drift, *Intell. Data Anal.* 3 (6) (1999) 453–474.
- [90] A. Narasimhamurthy, L.I. Kuncheva, A framework for generating data to simulate changing environments, in: *Proceedings of the 25th Conference on Proceedings of the 25th IASTED International Multi-Conference: Artificial Intelligence and Applications*, in: AIAP'07, ACTA Press, USA, 2007, pp. 384–389.
- [91] I. Katakis, G. Tsoumakas, I. Vlahavas, Tracking recurring contexts using ensemble classifiers: An application to email filtering, *Knowl. Inf. Syst.* 22 (3) (2010) 371–391.
- [92] F. Breve, L. Zhao, Semi-supervised learning with concept drift using particle dynamics applied to network intrusion detection data, in: *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, 2013, pp. 335–340.

- [93] H.S. Yazdi, A.G. Bafghi, et al., A drift aware adaptive method based on minimum uncertainty for anomaly detection in social networking, *Expert Syst. Appl.* 162 (2020) 113881.
- [94] M. Baena-Garcia, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, R. Morales-Bueno, Early drift detection method, in: *Fourth International Workshop on Knowledge Discovery from Data Streams*, Vol. 6, 2006, pp. 77–86.
- [95] K. Nishida, K. Yamauchi, Detecting concept drift using statistical testing, in: *International Conference on Discovery Science*, Springer, 2007, pp. 264–269.
- [96] R.S. Barros, D.R. Cabral, P.M. Gonçalves Jr., S.G. Santos, RDDM: Reactive drift detection method, *Expert Syst. Appl.* 90 (2017) 344–355.
- [97] I. Frias-Blanco, J. del Campo-Ávila, G. Ramos-Jimenez, R. Morales-Bueno, A. Ortiz-Diaz, Y. Caballero-Mota, Online and non-parametric drift detection methods based on Hoeffding's bounds, *IEEE Trans. Knowl. Data Eng.* 27 (3) (2014) 810–823.
- [98] W. Hoeffding, Probability inequalities for sums of bounded random variables, *J. Amer. Statist. Assoc.* 58 (301) (1963) 13–30.
- [99] A. Pesaranghader, H.L. Viktor, Fast hoeffding drift detection method for evolving data streams, in: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2016, pp. 96–111.
- [100] A. Pesaranghader, H. Viktor, E. Paquet, Reservoir of diverse adaptive learners and stacking fast hoeffding drift detection methods for evolving data streams, *Mach. Learn.* 107 (11) (2018) 1711–1743.
- [101] M.M.W. Yan, Accurate detecting concept drift in evolving data streams, *ICT Express* 6 (4) (2020) 332–338.
- [102] E. Lughofer, E. Weigl, W. Heidl, C. Eitzinger, T. Radauer, Recognizing input space and target concept drifts in data streams with scarcely labeled and unlabelled instances, *Inform. Sci.* 355 (2016) 127–151.
- [103] H. Mouss, D. Mouss, N. Mouss, L. Sefouhi, Test of page-hinckley, an approach for fault detection in an agro-alimentary production system, in: *2004 5th Asian Control Conference (IEEE Cat. No. 04EX904)*, Vol. 2, IEEE, 2004, pp. 815–818.
- [104] Y. Sakamoto, K.-I. Fukui, J. Gama, D. Nicklas, K. Moriyama, M. Numao, Concept drift detection with clustering via statistical change detection methods, in: *2015 Seventh International Conference on Knowledge and Systems Engineering (KSE)*, IEEE, 2015, pp. 37–42.
- [105] Z. Liu, C.K. Loo, M. Seera, Meta-cognitive recurrent recursive kernel OS-ELM for concept drift handling, *Appl. Soft Comput.* 75 (2019) 494–507.
- [106] N.A. Huynh, W.K. Ng, K. Ariyapala, Learning under concept drift with follow the regularized leader and adaptive decaying proximal, *Expert Syst. Appl.* 96 (2018) 49–63.
- [107] A. Andrzejak, J.B. Gomes, Parallel concept drift detection with online map-reduce, in: *2012 IEEE 12th International Conference on Data Mining Workshops*, IEEE, 2012, pp. 402–407.
- [108] S. Wang, L.L. Minku, D. Ghezzi, D. Caltabiano, P. Tino, X. Yao, Concept drift detection for online class imbalance learning, in: *The 2013 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2013, pp. 1–10.
- [109] H. Wang, Z. Abraham, Concept drift detection for streaming data, in: *2015 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2015, pp. 1–9.
- [110] S. Yu, Z. Abraham, H. Wang, M. Shah, Y. Wei, J.C. Principe, Concept drift detection and adaptation with hierarchical hypothesis testing, *J. Franklin Inst.* B 356 (5) (2019) 3187–3215.
- [111] D.K. Antwi, H.L. Viktor, N. Japkowicz, The PerfSim algorithm for concept drift detection in imbalanced data, in: *2012 IEEE 12th International Conference on Data Mining Workshops*, IEEE, 2012, pp. 619–628.
- [112] Y. Song, G. Zhang, H. Lu, J. Lu, A fuzzy drift correlation matrix for multiple data stream regression, in: *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, 2020, pp. 1–6.
- [113] S.-s. Zhang, J.-w. Liu, X. Zuo, Adaptive online incremental learning for evolving data streams, *Appl. Soft Comput.* 105 (2021) 107255.
- [114] R.T.M. Chikushi, R.S.M. de Barros, M.G.N.M. da Silva, B.I.F. Maciel, Using spectral entropy and bernoulli map to handle concept drift, *Expert Syst. Appl.* 167 (2021) 114114.
- [115] E. Oikarinen, H. Tiittanen, A. Henelius, K. Puolamäki, Detecting virtual concept drift of regressors without ground truth values, *Data Min. Knowl. Discov.* 35 (3) (2021) 726–747.
- [116] G.J. Ross, N.M. Adams, D.K. Tasoulis, D.J. Hand, Exponentially weighted moving average charts for detecting concept drift, *Pattern Recognit. Lett.* 33 (2) (2012) 191–198.
- [117] A.B. Yeh, R.N. Mcgrath, M.A. Sembower, Q. Shen, EWMA control charts for monitoring high-yield processes based on non-transformed observations, *Int. J. Prod. Res.* 46 (20) (2008) 5679–5699.
- [118] S. Disabato, M. Roveri, Learning convolutional neural networks in presence of concept drift, in: *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2019, pp. 1–8.
- [119] E.S. Page, Continuous inspection schemes, *Biometrika* 41 (1/2) (1954) 100–115.
- [120] S. Wang, L.L. Minku, AUC estimation and concept drift detection for imbalanced data streams with multiple classes, in: *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–8.
- [121] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, *Neurocomputing* 70 (1–3) (2006) 489–501.
- [122] N.-Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, *IEEE Trans. Neural Netw.* 17 (6) (2006) 1411–1423.
- [123] Z. Yang, S. Al-Dahidi, P. Baraldi, E. Zio, L. Montelatini, A novel concept drift detection method for incremental learning in nonstationary environments, *IEEE Trans. Neural Netw. Learn. Syst.* 31 (1) (2019) 309–320.
- [124] S. Xu, J. Wang, Dynamic extreme learning machine for data stream classification, *Neurocomputing* 238 (2017) 433–449.
- [125] B. Mirza, Z. Lin, Meta-cognitive online sequential extreme learning machine for imbalanced and concept-drifting data classification, *Neural Netw.* 80 (2016) 79–94.
- [126] A. Bifet, R. Gavalda, Learning from time-changing data with adaptive windowing, in: *Proceedings of the 2007 SIAM International Conference on Data Mining*, SIAM, 2007, pp. 443–448.
- [127] D.T.J. Huang, Y.S. Koh, G. Dobbie, R. Pears, Detecting volatility shift in data streams, in: *2014 IEEE International Conference on Data Mining*, IEEE, 2014, pp. 863–868.
- [128] R.S.M. de Barros, J.L.G. Hidalgo, D.R. de Lima Cabral, Wilcoxon rank sum test drift detector, *Neurocomputing* 275 (2018) 1954–1963.
- [129] F. Wilcoxon, Individual comparisons by ranking methods, in: *Breakthroughs in Statistics*, Springer, 1992, pp. 196–202.
- [130] D.R. de Lima Cabral, R.S.M. de Barros, Concept drift detection based on Fisher's Exact test, *Inform. Sci.* 442 (2018) 220–234.
- [131] R.A. Fisher, On the interpretation of χ^2 from contingency tables, and the calculation of P, *J. R. Stat. Soc.* 85 (1) (1922) 87–94.
- [132] J.L.G. Hidalgo, L.M.P. Mariño, R.S.M. de Barros, Cosine similarity drift detector, in: *International Conference on Artificial Neural Networks*, Springer, 2019, pp. 669–685.
- [133] O. Wu, Y.S. Koh, G. Dobbie, T. Lacombe, Nacre: Proactive recurrent concept drift detection in data streams, in: *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2021, pp. 1–8.
- [134] A. Pesaranghader, H.L. Viktor, E. Paquet, McDiarmid drift detection methods for evolving data streams, in: *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2018, pp. 1–9.
- [135] C. McDiarmid, et al., On the method of bounded differences, *Surv. Combin.* 141 (1) (1989) 148–188.
- [136] L. Du, Q. Song, X. Jia, Detecting concept drift: an information entropy based method using an adaptive sliding window, *Intell. Data Anal.* 18 (3) (2014) 337–364.
- [137] T.S. Sethi, M. Kantardzic, Don't pay for validation: Detecting drifts from unlabeled data using margin density, *Procedia Comput. Sci.* 53 (2015) 103–112.
- [138] A. Liu, G. Zhang, K. Wang, J. Lu, Fast switch Naïve Bayes to avoid redundant update for concept drift learning, in: *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–7.
- [139] A. Kolmogorov, Sulla determinazione empirica di una legge di distribuzione, *Inst. Ital. Attuari Giorn.* 4 (1933) 83–91.
- [140] I. Khamassi, M. Sayed-Mouchaweh, Drift detection and monitoring in non-stationary environments, in: *2014 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, IEEE, 2014, pp. 1–6.
- [141] I. Khamassi, M. Sayed-Mouchaweh, M. Hammami, K. Ghédira, Self-adaptive windowing approach for handling complex concept drift, *Cogn. Comput.* 7 (6) (2015) 772–790.
- [142] S. Liu, L. Lu, Y. Zhang, T. Xin, Y. Ji, R. Wang, Research on concept drift detection for decision tree algorithm in the stream of big data, in: *International Symposium on Parallel Architecture, Algorithm and Programming*, Springer, 2017, pp. 237–246.
- [143] B.I.F. Maciel, S.G.T.C. Santos, R.S.M. Barros, A lightweight concept drift detection ensemble, in: *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, 2015, pp. 1061–1068.
- [144] L. Du, Q. Song, L. Zhu, X. Zhu, A selective detector ensemble for concept drift detection, *Comput. J.* 58 (3) (2015) 457–471.
- [145] M. Woźniak, P. Ksieniewicz, B. Cyganek, K. Walkowiak, Ensembles of heterogeneous concept drift detectors-experimental study, in: *IFIP International Conference on Computer Information Systems and Industrial Management*, Springer, 2016, pp. 538–549.
- [146] N. Littlestone, M.K. Warmuth, The weighted majority algorithm, *Inform. and Comput.* 108 (2) (1994) 212–261.
- [147] W.N. Street, Y. Kim, A streaming ensemble algorithm (SEA) for large-scale classification, in: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 377–382.
- [148] H. Wang, W. Fan, P.S. Yu, J. Han, Mining concept-drifting data streams using ensemble classifiers, in: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003, pp. 226–235.

- [149] D. Brzeziński, J. Stefanowski, Accuracy updated ensemble for data streams with concept drift, in: *International Conference on Hybrid Artificial Intelligence Systems*, Springer, 2011, pp. 155–163.
- [150] D. Brzeziński, J. Stefanowski, Combining block-based and online methods in learning ensembles from concept drifting data streams, *Inform. Sci.* 265 (2014) 50–67.
- [151] J.-W. Liao, B.-R. Dai, An ensemble learning approach for concept drift, in: *2014 International Conference on Information Science & Applications (ICISA)*, IEEE, 2014, pp. 1–4.
- [152] D. Mejri, R. Khanchel, M. Limam, An ensemble method for concept drift in nonstationary environment, *J. Stat. Comput. Simul.* 83 (6) (2013) 1115–1128.
- [153] M.M. Idrees, L.L. Minku, F. Stahl, A. Badii, A heterogeneous online learning ensemble for non-stationary environments, *Knowl.-Based Syst.* 188 (2020) 104983.
- [154] P. Sidhu, M. Bhatia, A two ensemble system to handle concept drifting data streams: recurring dynamic weighted majority, *Int. J. Mach. Learn. Cybern.* 10 (3) (2019) 563–578.
- [155] R. Polikar, L. Upda, S.S. Upda, V. Honavar, Learn++: An incremental learning algorithm for supervised neural networks, *IEEE Trans. Syst. Man Cybern. C* 31 (4) (2001) 497–508.
- [156] G. Ditzler, R. Polikar, Incremental learning of concept drift from streaming imbalanced data, *IEEE Trans. Knowl. Data Eng.* 25 (10) (2012) 2283–2301.
- [157] S.G. Soares, R. Araújo, An on-line weighted ensemble of regressor models to handle concept drifts, *Eng. Appl. Artif. Intell.* 37 (2015) 392–406.
- [158] L.L. Minku, X. Yao, DDD: A new ensemble approach for dealing with concept drift, *IEEE Trans. Knowl. Data Eng.* 24 (4) (2011) 619–633.
- [159] P. Sidhu, M. Bhatia, An online ensembles approach for handling concept drift in data streams: diversified online ensembles detection, *Int. J. Mach. Learn. Cybern.* 6 (6) (2015) 883–909.
- [160] T. Museba, F. Nelwamondo, K. Ouahada, A. Akinola, Recurrent adaptive classifier ensemble for handling recurring concept drifts, *Appl. Comput. Intell. Soft Comput.* 2021 (2021).
- [161] O.A. Mahdi, E. Pardede, N. Ali, A hybrid block-based ensemble framework for the multi-class problem to react to different types of drifts, *Cluster Comput.* 24 (3) (2021) 2327–2340.
- [162] F. Pinagé, E.M. dos Santos, J. Gama, A drift detection method based on dynamic classifier selection, *Data Min. Knowl. Discov.* 34 (1) (2020) 50–74.
- [163] H.H. Ang, V. Gopalkrishnan, I. Zliobaite, M. Pechenizkiy, S.C. Hoi, Predictive handling of asynchronous concept drifts in distributed environments, *IEEE Trans. Knowl. Data Eng.* 25 (10) (2012) 2343–2355.
- [164] W. Liu, H. Zhang, Z. Ding, Q. Liu, C. Zhu, A comprehensive active learning method for multiclass imbalanced data streams with concept drift, *Knowl.-Based Syst.* 215 (2021) 106778.
- [165] K. Waiyamai, T. Kangkachit, B. Saengthongloun, T. Rakthanmanon, ACCD: Associative classification over concept-drifting data streams, in: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, Springer, 2014, pp. 78–90.
- [166] I. Khamassi, M. Sayed-Mouchaweh, M. Hammami, K. Ghédira, A new combination of diversity techniques in ensemble classifiers for handling complex concept drift, in: *Learning from Data Streams in Evolving Environments*, Springer, 2019, pp. 39–61.
- [167] T.S. Sethi, M. Kantardzic, Handling adversarial concept drift in streaming data, *Expert Syst. Appl.* 97 (2018) 18–40.
- [168] A. Haque, L. Khan, M. Baron, B. Thuraishingham, C. Aggarwal, Efficient handling of concept drift and concept evolution over stream data, in: *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, IEEE, 2016, pp. 481–492.
- [169] S. Khezri, J. Tanha, A. Ahmadi, A. Sharifi, A novel semi-supervised ensemble algorithm using a performance-based selection metric to non-stationary data streams, *Neurocomputing* 442 (2021) 125–145.
- [170] B. Mirza, Z. Lin, N. Liu, Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift, *Neurocomputing* 149 (2015) 316–329.
- [171] G.H. Oliveira, R.C. Cavalcante, G.G. Cabral, L.L. Minku, A.L. Oliveira, Time series forecasting in the presence of concept drift: A pso-based approach, in: *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, 2017, pp. 239–246.
- [172] Y. Xu, R. Xu, W. Yan, P. Ardis, Concept drift learning with alternating learners, in: *2017 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2017, pp. 2104–2111.
- [173] M. Dehghan, H. Beigy, P. ZareMoodi, A novel concept drift detection method in data streams using ensemble classifiers, *Intell. Data Anal.* 20 (6) (2016) 1329–1350.
- [174] S. Ren, B. Liao, W. Zhu, K. Li, Knowledge-maximized ensemble algorithm for different types of concept drift, *Inform. Sci.* 430 (2018) 261–281.
- [175] R. Anderson, Y.S. Koh, G. Dobbie, A. Bifet, Recurring concept meta-learning for evolving data streams, *Expert Syst. Appl.* 138 (2019) 112832.
- [176] B. Zhang, Y. Chen, Research on detection and integration classification based on concept drift of data stream, *EURASIP J. Wireless Commun. Networking* 2019 (1) (2019) 1–7.
- [177] R.C. Cavalcante, L.L. Minku, A.L. Oliveira, Fedd: Feature extraction for explicit concept drift detection in time series, in: *2016 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2016, pp. 740–747.
- [178] G. Ditzler, R. Polikar, Semi-supervised learning in nonstationary environments, in: *The 2011 International Joint Conference on Neural Networks*, IEEE, 2011, pp. 2741–2748.
- [179] T. Cerquitelli, S. Proto, F. Ventura, D. Apiletti, E. Baralis, Towards a real-time unsupervised estimation of predictive model degradation, in: *Proceedings of Real-Time Business Intelligence and Analytics*, 2019, pp. 1–6.
- [180] Y.-C. Ho, D.L. Pepyne, Simple explanation of the no-free-lunch theorem and its implications, *J. Optim. Theory Appl.* 115 (3) (2002) 549–570.
- [181] V. Buhrmester, D. Münch, M. Arens, Analysis of explainers of black box deep neural networks for computer vision: A survey, 2019, arXiv preprint arXiv:1911.12116.
- [182] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, B.Y. Zhao, Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, in: *2019 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2019, pp. 707–723.
- [183] J. Lu, A. Liu, Y. Song, G. Zhang, Data-driven decision support under concept drift in streamed big data, *Complex Intell. Syst.* 6 (1) (2020) 157–163.
- [184] K. Wang, J. Lu, A. Liu, G. Zhang, L. Xiong, Evolving gradient boost: A pruning scheme based on loss improvement ratio for learning under concept drift, *IEEE Trans. Cybern.* (2021).
- [185] I.H. Sarker, Machine learning: Algorithms, real-world applications and research directions, *SN Comput. Sci.* 2 (3) (2021) 1–21.
- [186] R.S.M. Barros, S.G.T.C. Santos, A large-scale comparison of concept drift detectors, *Inform. Sci.* 451 (2018) 348–370.
- [187] J.L. Lobo, J. Del Ser, A. Bifet, N. Kasabov, Spiking neural networks and online learning: An overview and perspectives, *Neural Netw.* 121 (2020) 88–100.
- [188] Y. Cao, H. Peng, J. Wu, Y. Dou, J. Li, P.S. Yu, Knowledge-preserving incremental social event detection via heterogeneous GNNs, in: *Proceedings of the Web Conference 2021*, 2021, pp. 3383–3395.
- [189] X. Bai, X. Wang, X. Liu, Q. Liu, J. Song, N. Sebe, B. Kim, Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments, *Pattern Recognit.* (2021) 108102.
- [190] Z. Chen, Y. Bei, C. Rudin, Concept whitening for interpretable image recognition, *Nat. Mach. Intell.* 2 (12) (2020) 772–782.