# On the Hardness and Necessity of Supervised Concept Drift Detection*

Fabian Hinder[a], Valerie Vaquet[b], Johannes Brinkrolf[c], and Barbara Hammer[d]
*CITEC, Bielefeld University, Inspiration 1, Bielefeld, Germany*

Keywords: Concept Drift, Stream Learning, Drift Detection, No Free Lunch.

Abstract: The notion of concept drift refers to the phenomenon that the distribution generating the observed data changes over time. If drift is present, machine learning models can become inaccurate and need adjustment. Many technologies for learning with drift rely on the interleaved test-train error to detect drift and trigger model updates. This type of drift detection is also used for monitoring systems aiming to detect anomalies. In this work, we analyze the relationship between concept drift and change of loss on a theoretical level. We focus on the sensitivity, specificity, and localization of change points in drift detection, putting an emphasize on the detection of real concept drift. With this focus, we compare the supervised and unsupervised setups which are already studied in the literature. We show that, unlike the unsupervised case, there is no universal supervised drift detector and that the assumed correlation between model loss and concept drift is invalid. We support our theoretical findings with empirical evidence for a combination of different models and data sets. We find that many state-of-the-art supervised drift detection methods suffer from insufficient sensitivity and specificity, and that unsupervised drift detection methods are a promising addition to existing supervised approaches.

## 1 INTRODUCTION

The world that surrounds us is undergoing constant change, which also affects the increasing amount of data sources available. Those changes – referred to as concept drift – frequently occur when data is collected over time, e.g., in social media, sensor networks, IoT devices, etc., and are induced by several causes such as seasonal changes, changed demands of individual costumers, aging, or failure of sensors, etc. Drift in the data usually requires some actions being taken to ensure that systems are running smoothly. These can be either actions taken by a person or by the learning algorithm (Ditzler et al., 2015).

Understanding the nature and underlying structure of drift is important as it allows the user to make informed decisions (Webb et al., 2017) and technical systems to perform desirable corrections (Vaquet et al., 2022). Depending on the context, different actions have to be taken. There are two main problem setups: Autonomously running systems need to robustly solve a given task in the presence of drift. In this setting, the learning algorithm of online learners needs knowledge about the drift to update the model in a reasonable way (Ditzler et al., 2015). In contrast, in system monitoring, the drift itself is of interest as it might indicate that certain actions have to be taken. Examples of such settings are cyber-security, where a drift indicates a potential attack, and the monitoring of critical infrastructures such as electric grids or water distribution networks, where drift indicates leakages or other failures (Eliades and Polycarpou, 2010). While system monitoring can minimize the damage caused by a malfunctioning technical system, and reduce wastage of resources, applying adaptive online learners can increase revenue in case of changing consumer behaviors and is key to robot navigation and autonomous driving (Losing et al., 2015).

Although both problem setups are very different, the majority of approaches rely on (supervised) drift detection, where the drift is detected by analyzing changes in the loss of online models. This is an intuitive step in the setting of (supervised) online learning as the goal is to minimize the interleaved test-train error by triggering model updates when drift is detected. In the monitoring scenario, detecting drift by referring to a stream learning setup is a commonly used surrogate for the actual problem.

[a] https://orcid.org/0000-0002-1199-4085
[b] https://orcid.org/0000-0001-7659-857X
[c] https://orcid.org/0000-0002-0032-7623
[d] https://orcid.org/0000-0002-0935-5591
*We gratefully acknowledge funding by the BMBF TiM, grant number 05M20PBA.



Hinder, F., Vaquet, V., Brinkrolf, J. and Hammer, B.
On the Hardness and Necessity of Supervised Concept Drift Detection.
DOI: 10.5220/0011797500003411
In *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2023)*, pages 164-175
ISBN: 978-989-758-626-2; ISSN: 2184-4313

Considering stream learning in the presence of concept drift, technologies commonly rely on windowing techniques and adapt the model based on the characteristics of the data in an observed time window. Such methods rely on non-parametric methods and ensemble technologies for (mostly supervised) online models. Active methods explicitly detect drift, usually referring to change of loss, and trigger model adaptation this way, while passive methods continuously adjust the model (Ditzler et al., 2015), hybrid approaches (Raab et al., 2019) combine both approaches by continuously adjusting the model unless drift is detected and a new model is trained.

For many problems the precise pinpointing of the timepoint of the drift event is mandatory to ensure an optimal usage of the provided data: Only if we know which samples were collected after the drift happened, we can perform retraining or analysis in a consistent way with respect to the current distribution. In addition to a precise pinpointing in time, a distinction between virtual and real drift, i.e., non-stationarity of the marginal distribution only or also the posterior, can help to understand the dynamics of the drift and inform consecutive steps.

The detection of drift, especially real drift is generally considered to be a hard problem (Hu et al., 2020). Although many attempts were made to tackle the problem of constructing a general purpose detector for real drift, it is still considered to be widely unsolved. Recently theoretical results regarding the solubility were published (Hinder et al., 2020, 2022) validating a large class of common drift detection schemes from a theoretical perspective. However, those results focus on drift detection in an unsupervised scenario only. To the best of our knowledge, comparable results do not exist for drift detection in a supervised setup, i.e., for the detection of real drift.

The purpose of this contribution is to deepen the understanding of drift detection from a theoretical point of view by analyzing the interconnection between concept drift and learning algorithms. More precisely, we consider the commonly assumed necessity of concept drift detection for the validity of stream learning algorithms (Gonçalves Jr et al., 2014; Gama et al., 2004, 2014) and, conversely, the applicability of commonly applied drift detection schemes from stream learning to concept drift detection as a statistical problem (Eliades and Polycarpou, 2010), as is common practice for monitoring problems. In particular, we analyze the implications of our results for monitoring setups and stream learning tasks. This includes advice for practical applications and an impossibility result regarding universal supervised drift detection.

As a result, we can answer the following questions in the context of supervised drift detection, i.e., virtual and real drift, which suggest several important corollaries regarding the possibility and interconnection of supervised and unsupervised drift detection and the connection between system monitoring and stream learning:

1. Can we detect the drift (sensitivity)?
   *Only if we do not miss drifts we can ensure robust monitoring and reliable model updates.*
2. Can we be sure about the detection (specificity)?
   *False alarms can be costly in monitoring applications and trigger unwanted updates in online learning which might be harmful for the model's performance.*
3. Can we determine the timepoint of the drift (localization precision)?
   *Large detection delays pose risks in monitoring tasks and delay the update of online learners.*

This paper is organized as follows: First (Section 2) we recall the basic notions of statistical learning theory and concept drift followed by reviewing the existing literature, mainly focusing on drift detection. We proceed with a theoretical analysis starting with a precise mathematical formalization of the notions of real and virtual drift and the analysis thereof (Section 3.2), followed by an analysis of the suitability of stream learners for drift detection (Section 3.3). Afterward, we empirically quantify the theoretical findings (Section 4) and conclude with a summary (Section 5).

# 2 PROBLEM SETUP, NOTATION, AND RELATED WORK

We make use of the formal framework for concept drift as introduced by Hinder et al. (2020, 2019) as well as classical statistical learning theory, e.g., as presented by Shalev-Shwartz and Ben-David (2014). In this section, we recall the basic notions of both subjects followed by a summary of the related work on concept drift detection schemes focusing on a high-level point of view.

## 2.1 Basic Notions of Statistical Learning Theory

In classical learning theory, one considers a hypothesis class $\mathcal{H}$, e.g., a set of functions from $\mathbb{R}^d$ to $\mathbb{R}$, together with a non-negative loss function $\ell : \mathcal{H} \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}_{\geq 0}$ that is used to evaluate how well a model $h$ matches an observation $(x,y) \in \mathcal{X} \times \mathcal{Y}$ by assigning an error $\ell(h,(x,y))$. We will refer to $\mathcal{X}$ as the data space and $\mathcal{Y}$ as the label space. For a given distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$ we consider $\mathcal{X}$- and $\mathcal{Y}$-valued random variables $X$ and $Y$, $(X,Y) \sim \mathcal{D}$, and assign the loss $\mathcal{L}_{\mathcal{D}}(h) = \mathbb{E}[\ell(h,(X,Y))]$ to a model $h \in \mathcal{H}$. Using a data sample $S \in \cup_{N\in\mathbb{N}}(\mathcal{X}\times\mathcal{Y})^N$ consisting of i.i.d. random variables $S = ((X_1,Y_1),\dots,(X_n,Y_n))$ distributed according to $\mathcal{D}$, we can approximate $\mathcal{L}_{\mathcal{D}}(h)$ using the empirical loss $\mathcal{L}_S(h) = \frac{1}{n}\sum_{i=1}^n \ell(h,(X_i,Y_i))$, which converges to $\mathcal{L}_{\mathcal{D}}(h)$ almost surely. Popular loss functions are the mean squared error $\ell(h,(x,y)) = (h(x)-y)^2$, cross-entropy $\ell(h,(x,y)) = -\sum_{i=1}^n \mathbf{1}[y=i]\log(h(i \mid x))$, or the 0-1-loss $\ell(h,(x,y)) = \mathbf{1}[h(x) \neq y]$.

In machine learning, training a model often refers to minimizing the loss $\mathcal{L}_{\mathcal{D}}(h)$ using the empirical loss $\mathcal{L}_S(h)$ as a proxy. A learning algorithm $A$, such as gradient descent schemes, selects a model $h$ given a sample $S$, i.e., $A : \cup_N(\mathcal{X}\times\mathcal{Y})^N \to \mathcal{H}$. Classical learning theory investigates under which circumstances $A$ is consistent, that is, it selects a good model with high probability: $\mathcal{L}_{\mathcal{D}}(A(S)) \to \inf_{h^*\in\mathcal{H}} \mathcal{L}_{\mathcal{D}}(h^*)$ as $|S| \to \infty$ in probability. Since the model $A(S)$ is biased towards the loss $\mathcal{L}_S$ due to training, classical approaches aim for uniform bounds $\sup_{h\in\mathcal{H}}|\mathcal{L}_S(h) - \mathcal{L}_{\mathcal{D}}(h)| \to 0$ as $|S| \to \infty$ in probability.

## 2.2 A Statistical Framework for Concept Drift

The classical setup of learning theory assumes a time-invariant distribution $\mathcal{D}$ for all $(X_i,Y_i)$. This assumption is violated in many real-world applications, in particular, when learning on data streams. Therefore, we incorporate time into our considerations by means of an index set $\mathcal{T}$, representing time, and a collection of (possibly different) distributions $\mathcal{D}_t$ on $\mathcal{X}\times\mathcal{Y}$, indexed over $\mathcal{T}$ (Gama et al., 2014). In particular, the model $h$ and its loss also become time-dependent. It is possible to extend this setup to a general statistical interdependence of data and time via a distribution $\mathcal{D}$ on $\mathcal{T}\times(\mathcal{X}\times\mathcal{Y})$ which decomposes into a distribution $\mathbb{P}_T$ on $\mathcal{T}$ and the conditional distributions $\mathcal{D}_t$ on $\mathcal{X}\times\mathcal{Y}$, the tuple $(\mathcal{D}_t,\mathbb{P}_T)$ is called a *(supervised) drift process* (Hinder et al., 2020, 2019). Our main example is binary classification on a time interval, i.e., $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{0,1\}$, and $\mathcal{T} = [0,1]$.

Drift refers to the fact that $\mathcal{D}_t$ varies for different timepoints, i.e., $\{(t_0,t_1) \in \mathcal{T}^2 : \mathcal{D}_{t_0} \neq \mathcal{D}_{t_1}\}$ has measure larger zero w.r.t $\mathbb{P}_T^2$ (Hinder et al., 2020). One further distinguishes a change of the posterior $\mathcal{D}_t(Y \mid X)$, referred to as *real drift*, and of the marginal $\mathcal{D}_t(X)$, referred to as *virtual drift*. One of the key findings of Hinder et al. (2020) is a unique characterization of the presence of drift by the property of statistical dependency of time $T$ and data $(X,Y)$ if a time-enriched representation of the data $(T,X,Y) \sim \mathcal{D}$ is considered. The task of determining whether or not there is drift during a time period is called *drift detection*. Following the terminology in learning tasks, we will refer to the detection of real drift, i.e., of the posterior $\mathcal{D}_t(Y \mid X)$ only, as *supervised* and (virtual) drift, i.e., in the marginal $\mathcal{D}_t(X)$ or the joint distribution $\mathcal{D}_t(X,Y)$ (mathematically those are the same), as *unsupervised* drift detection. We say that a drift detector is *universal* if it is capable of raising correct alarms with a high probability independent of the distribution(s) in the stream, assuming a sufficient amount of data is provided.

In this work, we will consider data drawn from a single drift process, thus we will make use of the following short-hand notation $\mathcal{L}_t(h) := \mathcal{L}_{\mathcal{D}_t}(h)$ for a timepoint $t \in \mathcal{T}$ and $\mathcal{L}(h) := \mathcal{L}_{\mathcal{D}(X,Y)}(h)$ is the loss on the entire stream.

## 2.3 Related Work and Existing Methods

There is only little work on learning theory for drift detection. What is known about learning theory in the context of drift is concerned with learning guarantees in stream learning, or learning of statistical processes and time series analysis (Mohri and Muñoz Medina, 2012; Hanneke et al., 2015). To the best of our knowledge, there is no other strain of work that deals with the question at hand in comparable generality or setup.

In the following, we will give a survey of the literature on drift detection methods from a high-level point of view. As pointed out by Lu et al. (2018); Hinder et al. (2022), basically all drift detection methods, independent of whether they are applied in the stream learning or monitoring setup, supervised or unsupervised, are essentially based on comparing time windows. The respective samples might be stored directly or implicitly in a descriptive statistic or machine learning model (Hinder et al., 2022). Also, such steps can be performed several times, e.g., ADWIN (Bifet and Gavaldà, 2007) first stores a reference window in a model which is then applied to the incoming data resulting in a stream of losses which is then analyzed using an auto-cut statistic on a sliding window. There are also hierarchical and ensemble approaches, that combine several drift detectors. This leads to a large variety of methods, but it does not change principle criticism, as such methods inherit the principle strengths and weaknesses of the internally used methods.

Formally, one can consider drift detectors as a kind of statistical test that aims to differentiate between the null hypothesis "for all timepoints $t$ and $s$ we have $\mathcal{D}_t = \mathcal{D}_s$" and the alternative "we may find timepoints $t$ and $s$ with $\mathcal{D}_t \neq \mathcal{D}_s$". It can be shown that it is sufficient to consider the distribution before and after a timepoint $t$, for any distribution and type of drift, as is implicitly assumed by many drift detectors (Hinder et al., 2019). In this sense, unsupervised drift detection can be considered as a sequence of two sample tests applied to a stream. This implies that we need very strong statistics in order to deal with the multi-testing problem. Supervised drift detection on the other hand is similar to conditional independence testing, as we will see in Theorem 1. This implies that every supervised drift detector induces an unsupervised drift detector, but not the other way around.

One way to obtain strong statistics for both scenarios is to rely on machine learning models and their loss as surrogates, as already mentioned above. The main advantage of this approach is that we only need to check for the change in the mean of a one-dimensional random variable, i.e., the empirical loss. Furthermore, we can apply such approaches in the supervised, e.g., by using the accuracy of a classifier (Page, 1954; Bifet and Gavaldà, 2007; Gama et al., 2004; Frías-Blanco et al., 2015; Baena-García et al., 2006; Gonçalves Jr et al., 2014), and unsupervised setup, e.g., by using the negative log-likelihood of a density estimator or virtual classifiers (Gretton et al., 2006; Gözüaçık et al., 2019; Bu et al., 2016). Thus, such methods can easily be adjusted to the data by using model selection. Furthermore, if we are actually interested in using this model, as is the case for active methods in stream learning, the relevance of this approach becomes obvious.

# 3 A THEORETICAL ANALYSIS

We will now consider real drift and supervised drift detection from a theoretical perspective. We will start by recalling the already established knowledge of unsupervised drift detection. Then, we provide a precise formal definition of the notions of *real* and *virtual drift*. We proceed by analyzing those to derive equal formulations comparable to those presented by Hinder et al. (2020). In particular, we analyze the interconnection of real drift and drift of the joint distribution and thereby supervised and unsupervised drift detection (Corollary 1) which will allow us to answer Questions 3. We then analyze the interconnection of drift and learning: We show that the effect of drift heavily depends on the precise setup including both the drift and the hypothesis class – which also answers Question 1 and 2. In particular, we will show that for many common learning models, the connection between model loss, optimal model, and type of drift is rather vague (Theorem 3). This implies that many supervised drift detection algorithms that rely on model loss to detect drift are only well suited if the hypothesis class and the drift match. As a consequence, we can conclude that there is no general purpose, i.e., universal, supervised drift detector.

## 3.1 Unsupervised Drift Detection

Let us first recapitulate the main results from the unsupervised drift detection scenario which has affirmatively been solved in the literature.

**Definition 1.** An *unsupervised drift detector* is a decision algorithm on data-time-pairs of any sample size $n$, i.e., $A : \cup_{N\in\mathbb{N}}(\mathcal{T}\times\mathcal{X})^N \to \{0,1\}$. A drift detector $A$ is *valid* on a set of drift processes $\mathfrak{D}$, iff

$$\limsup_{n\to\infty} \sup_{\substack{\mathcal{D}_t\in\mathfrak{D} \\ \mathcal{D}_t \text{ has no drift}}} \mathbb{P}_{S\sim\mathcal{D}^n}[A(S)=1] < \inf_{\substack{\mathcal{D}_t\in\mathfrak{D} \\ \mathcal{D}_t \text{ has drift}}} \limsup_{n\to\infty} \mathbb{P}_{S\sim\mathcal{D}^n}[A(S)=1].$$

We say that $A$ is *universal* if it is valid for all possible streams, i.e., $\mathfrak{D}$ is the set of all drift processes.

It was shown that unsupervised drift detection, without a specification of a change point, is equivalent to independence testing (Hinder et al., 2020) for which well-known and good performing tests exist – this answers Questions 1 and 2 positive in a general setup. If we also want to know the change points, we can consider the problem as a kind of bi-clustering problem which is learnable in many cases (Hinder et al., 2022). This approach is very general and one can turn nearly every binary classifier into a drift detector that, as sample size goes to infinity, has arbitrary high sensitivity, specificity, and precision regarding the change point localization – also answering Question 3 positive in a general setup. To summarize: There exists a universal unsupervised drift detector and we can answer all three research questions positive.

## 3.2 Supervised Concept Drift Detection as a Statistical Problem

To consider the case of supervised drift detection we first need a precise formalization of real and virtual drift: The notion of virtual drift, a change of the distribution of $X$, is just drift of the marginal distribution and thus can easily be adapted in the framework of Hinder et al. (2020). To extend the notion of real drift, i.e., the fact that the classification rule $x \mapsto y$ changes over time, to the probabilistic setup, we have to be a bit more careful as

$$\mathcal{D}_t(Y \mid X=x) \neq \mathcal{D}_s(Y \mid X=x)$$

only implies real drift if we can actually observe $x$ at both timepoints $t$ and $s$, i.e., $\mathcal{D}_t(X=x), \mathcal{D}_s(X=x) > 0$, otherwise this statement is meaningless. Furthermore, it is not clear how to compare the two distributions, as conditioning on a non-finite space requires heavy mathematical machinery and will in general not result in a probability measure, let alone a unique one. Instead, we will focus on derived statistics, i.e., bounded functions $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, and whether they are affected by the drift. One of the main examples of such statistics that is of interest for us is the loss of a given, fixed model, i.e., $\ell(h,\cdot) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, which essentially considers the change of model loss if confronted with drift. To obtain a quantitative measure of drift from $f$, we proceed as follows: First, we consider the conditional expectation given data and time but no label which is a function of data and time, i.e., we consider

$$\mathbb{E}[f(X,Y) \mid X,T] : \mathcal{X} \times \mathcal{T} \to \mathbb{R}.$$

If the distribution of $Y \mid X$ does not change with time then this function is $T$ invariant. The idea is now to measure this invariance: This can be done by comparing $\mathbb{E}[f(X,Y) \mid X,T]$ to the best possible approximation of it that depends on $X$ only. Using the $L^2$-norm we obtain

$$\inf_{g \in L^2(\mathcal{D}(X))} \|\mathbb{E}[f(X,Y) \mid X,T] - g(X)\|_{L^2(\mathcal{D}(T,X))},$$

which is known to be just the conditional expectation given $X$ , i.e., $g(X) = \mathbb{E}[\mathbb{E}[f(X,Y) \mid X,T] \mid X] = \mathbb{E}[f(X,Y) \mid X]$. This allows us to express the last equation in terms of conditional variation:

$$\mathbb{E}_{X\sim\mathcal{D}(X)}[\mathrm{Var}(\mathbb{E}[f(X,Y) \mid X,T] \mid X)].$$

This is zero if and only if we can estimate the statistic without referring to $T$, i.e., if it is $T$ invariant, and thus leads to the following definition:

**Definition 2.** Let $(\mathcal{D}_t, \mathbb{P}_T)$ be a supervised drift process, i.e., a Markov kernel $\mathcal{D}_t$ from $\mathcal{T}$ to $\mathcal{X} \times \mathcal{Y}$ together with a probability measure $\mathbb{P}_T$ on $\mathcal{T}$. We say that $\mathcal{D}_t$ has *virtual drift* iff the marginal on $\mathcal{X}$ has drift in the usual sense, i.e., if $(\mathcal{D}_t(X), \mathbb{P}_T)$ has drift. We say that $\mathcal{D}_t$ has *real drift* iff there is a time varying statistic, i.e., i.e.,

$$\exists f : \mathbb{E}[\mathrm{Var}(\mathbb{E}[f(X,Y) \mid X,T] \mid X)] > 0.$$

We say that $\mathcal{D}_t$ has *virtual/real drift only* iff it has virtual/real drift but no real/virtual drift.

The notions of *supervised drift detector*, *validity*, and *universality* are analogous to the unsupervised case except that we consider real drift in the definition of validity.

Note that this is a slight variation from the usual nomenclature where real/virtual drift refers to the situation we call real/virtual drift only. Indeed, if $\mathcal{D}_t$ has no virtual drift, then our definition coincides with the usual definition (see Lu et al., 2018). However, to analyze the change of the posterior $Y \mid X$ and the distribution of $X$ at the same time, such a definition is necessary.

As our definition of real drift is rather complicated, we will continue by providing equivalent formalization comparable to Hinder et al. (2020):

**Theorem 1.** *Let $\mathcal{D}_t$ be a supervised drift process. The following are equivalent*

1. *$\mathcal{D}_t$ has no real drift*
2. *For any statistic, points for which it varies in time are not observed, i.e., $\sup_f \mathcal{D}_X[\mathrm{Var}(\mathbb{E}[f(X,Y) \mid X,T] \mid X) \neq 0] = 0$.*
3. *Every statistic admits a time-invariant prediction, i.e., for all bounded $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ there exists a $g : \mathcal{X} \to \mathbb{R}$ such that $\mathbb{E}[f(X,Y) \mid X,T] = g(X)$.*
4. *There is no (conditional) dependency drift (Hinder et al., 2020): If $(X,Y,T) \sim \mathcal{D}$ then $T$ and $Y$ are conditionally independent given $X$, i.e., $Y \perp\!\!\!\perp T \mid X$.*

*Proof.* **"1 ⇒ 4":** Let $A \subset \mathcal{Y}$ and consider $f(x,y) = \mathbb{I}_A(y)$. Then we have

$$\begin{aligned} 0 &\geq \mathbb{E}[\mathrm{Var}(\mathbb{E}[f(X,Y) \mid X,T] \mid X)] \\ &= \|\mathbb{P}_{Y|X,T}(A) - \mathbb{P}_{Y|X}(A)\|^2_{L^2(\mathcal{D}(T,X))}. \end{aligned}$$

This is the case for all $A$ if and only if $Y \perp\!\!\!\perp T \mid X$.

**"4 ⇒ 3":** If $Y \perp\!\!\!\perp T \mid X$ then $\mathbb{E}[f(X,Y) \mid X,T] = \mathbb{E}[f(X,Y) \mid X]$, so we may choose $g(X) = \mathbb{E}[f(X,Y) \mid X]$.

**"3 ⇒ 1":** Since $\mathbb{E}[f(X,Y) \mid X]$ is the (according to $L^2$) best possible approximation of $\mathbb{E}[f(X,Y) \mid X,T]$ that depends on $X$ only and $\mathbb{E}[f(X,Y) \mid X,T] = g(X)$, which depends on $X$ only, we have $\mathbb{E}[f(X,Y) \mid X,T] = g(X) = \mathbb{E}[f(X,Y) \mid X]$ and therefore the variance is 0.

**"2 ⇔ 1":** Consider $V_f(x) = \mathrm{Var}(\mathbb{E}[f(X,Y) \mid X,T] \mid X = x)$ which is a non-negative, measurable function of $x$. Then it holds $\mathbb{P}[V_f(X) > 0] = 0 \Leftrightarrow \mathbb{E}[V_f(X)] = 0$ and the supremum is 0 if and only if all $f$ result in 0. □

Notice that the notion of virtual and real drift are together equivalent to drift on the joint distribution of $(X,Y)$. Formally we have:

**Corollary 1.** *A supervised drift process $\mathcal{D}_t$ has drift if and only if $\mathcal{D}_t$ has virtual or real drift.*

*Proof.* We show that there is no drift if and only if there is no virtual and no real drift. Using Theorem 1 and (Hinder et al., 2020, Theorem 3) it follows by contraction that

$$\underbrace{Y \perp\!\!\!\perp T \mid X}_{\text{no real drift}} \text{ and } \underbrace{X \perp\!\!\!\perp T}_{\text{no virtual drift}} \Rightarrow \underbrace{X, Y \perp\!\!\!\perp T}_{\text{no (dependency) drift}} .$$

For the other direction, by weak union and decomposition, it follows

$$\underbrace{X, Y \perp\!\!\!\perp T}_{\text{no (dependency) drift}} \Rightarrow \underbrace{Y \perp\!\!\!\perp T \mid X}_{\text{no real drift}} \quad \text{and}$$
$$\underbrace{X, Y \perp\!\!\!\perp T}_{\text{no (dependency) drift}} \Rightarrow \underbrace{X \perp\!\!\!\perp T}_{\text{no virtual drift}} .$$

Which completes the proof. □

This corollary is important as it answers our Question 3, i.e., change point localization, by allowing us to carry over the result from unsupervised drift detection to the supervised setup by considering the joint distribution. The main challenge is thus to assure specificity in the supervised setup.

Practically, this means that we can always make use of unsupervised drift detection to identify potential change points and then double check using a suitable test. It is therefore reasonable that we focus on the case where $\mathcal{T}$ is finite.

Checking for real drift in such a setup can be done by comparing the optimal model that respects the additional temporal information, i.e., may change over time, with the time-independent one. We will refer to the types of drift that affect models as *model drift*. This consideration leads to the following setup which relates the loss of optimal models and drift:

**Definition 3.** Let $\mathcal{T}$ be finite, $\mathcal{D}_t$ a supervised drift process, and $\mathcal{H}$ a hypothesis class with loss $\ell$. We call the following the *detection equation*:

$$\mathbb{E}\left[\inf_{h^* \in \mathcal{H}} \mathcal{L}_T(h^*)\right] \leq \inf_{h^* \in \mathcal{H}} \mathcal{L}(h^*).$$

Iff the inequality is strict, $\mathcal{H}$ has *model drift*.

The detection equation can be motivated by Theorem 1.3 which implies that we can replace the time-dependent model with a time-independent model if and only if there is no real drift, assuming a sufficiently large hypothesis class. It can be shown that there is actually a deep connection between model drift and real drift, as is shown by the following result:

**Theorem 2.** *Let $\mathcal{Y} = \{0,1\}$, $\mathcal{X} = \mathbb{R}^d$, $\mathcal{T}$ be finite, and $\mathcal{H}$ be universal, i.e., $\mathcal{H}$ are measurable, bounded functions $h : \mathcal{X} \to \mathbb{R}$ and for every continuous $f : \mathcal{X} \to \mathbb{R}$ with compact support and $\varepsilon > 0$, there exists a $h^* \in \mathcal{H}$ such that $\sup_{x \in \mathcal{X}} |h^*(x) - f(x)| < \varepsilon$. Then there is real drift if and only if $\mathcal{H}$ with MSE-loss has model drift.*

*Proof.* Since the MSE decomposes into $\mathbb{E}[(h(X) - \mathbb{E}[Y \mid X])^2]$ and $\mathbb{E}[\mathrm{Var}(Y \mid X)]$ and the second part does not depend on $h$, we can subtract it from both sides of the detection equation, which then becomes

$$\mathbb{E}\left[\inf_{h^* \in \mathcal{H}} \mathbb{E}[(h^*(X) - \mathbb{E}[Y \mid X, T])^2 \mid T]\right] \leq \inf_{h^* \in \mathcal{H}} \mathbb{E}[(h^*(X) - \mathbb{E}[Y \mid X, T])^2].$$

As $\mathcal{H}$ is dense in $L^2$ the left-hand side is 0 and, using Theorem 1.3, the right-hand side is 0 if and only if there is no real drift. □

This result is promising as it shows that, at least in theory, supervised drift detection is possible. In the next section, we will consider the problem from a statistical point of view, i.e., in the case where we are provided with a finite amount of data only.

## 3.3 On the Hardness of Supervised Drift Detection

So far we have seen that many results from the unsupervised statistical problem setup carry over to supervised drift detection and model drift. In this section, we will consider the mismatch between the setups, particularly, the mismatch between model drift and supervised drift detection. To do so, we consider an unspecified statistical test that probes for real drift by checking for model drift as a proxy, i.e., it uses model loss as a preprocessing. According to the two types of error of the statistical test, we consider two types of mismatches:

**Definition 4.** Let $\mathcal{D}_t$ be a drift process, $\mathcal{H}$ be a hypothesis class with a loss function $\ell$.

- *Mismatch of the first kind:* $\mathcal{D}_t$ has real drift, but no model drift.
- *Mismatch of the second kind:* $\mathcal{D}_t$ has no real drift, but model drift.

Notice that this definition reflects our research questions in a formal way: If the sensitivity is high then a mismatch of the first kind is unlikely. If the specificity is high then a mismatch of the second kind is unlikely. As we have already answered the question of localization of change points using Corollary 1 to reduce the supervised problem to the unsupervised (Hinder et al., 2022), we will now focus on considering the occurrence of mismatches.

In Theorem 2, we required that $\mathcal{H}$ is a rather large hypothesis class. However, we cannot omit such an assumption due to the fact that if the hypothesis class is "small" it is certain to fail:

**Theorem 3** (No free lunch). *Let $\mathcal{H}$ be a hypothesis class of binary classifiers with 0-1-loss of VC-dimension $d < \infty$. For any $S \subset \mathcal{X}$ of size larger than*

$4(d+1)$ *there is a mismatch of the first kind and if* $d < \log_2 |\mathcal{H}|$ *there is a* $S \subset \mathcal{X}$ *of size not lager than* $d+1$ *with mismatch of the second kind, i.e., for any choice of* $\mathcal{T}$ *and* $\mathbb{P}_T$ *(that is not concentrated on a single point) there exists a drift process* $\mathcal{D}_t$ *on S without noise, i.e.,* $\mathcal{D}_t(Y=1 \mid X) \in \{0,1\}$ *and* $\mathcal{D}_t(X \in S) = 1$*, for which the mismatch occurs.*

*Proof.* **Mismatch of the First Kind:** Let $S \subset \mathcal{X}$ be a set of size $m > d+1$. By Sauer's lemma we have $|\mathcal{H}_{|S}| \leq (em/d)^d$, here $\mathcal{H}_{|S}$ denotes the restriction of $\mathcal{H}$ onto $S$. Furthermore, there exist $2^m$ labelings $f : S \to \{0,1\}$ but the loss of any $h \in \mathcal{H}$ can take on at most $m+1$ values, i.e., $\{\sum_{x \in S} \mathbf{1}[h(x) \neq f(x)] \mid h \in \mathcal{H}\} \subset \{0,1,\ldots,m\}$. Thus, there exist at most $(em/d)^d(m+1)$ different combinations of model and loss. Therefore, if we assign each labeling to those model(s) with the respective smallest error we end up with assigning two labelings $f_1 \neq f_2$ to the same model $h$ having the same loss, as the number of labelings grows faster with respect to $m$ than the number of model-loss combinations. Furthermore, $h$ is the best possible choice for $f_1$ and $f_2$ among all other models. By using $f_1$ to label the first and $f_2$ to label the second part of the stream, we end up with a stream with real drift but neither the optimal model nor its loss change.

Let us now compute $m$. In order to apply Sauer's lemma and the counting argument we need to have

$$\frac{2^m}{\left(\frac{em}{d}\right)^d (m+1)} > 1 \qquad m > d+1,$$

where the $> 1$ is sufficient because counts can only take integers, so at least one instance must take the value 2. Since $m$ has to grow at least linearly with respect to $d$ in order to fulfill the second condition, we will first determine a factor $v$ such that $m = vd$ is sufficient to fulfill the second property in the limit $d \to \infty$. Since the first criterion takes on non-negative values only, we can take the logarithm on both sides

$$(v\log(2) - \log(v) - 1) - \underbrace{\log(1+dv)/d}_{\xrightarrow{d\to\infty} 0} > 0.$$

Since $v\log(2)$ grows faster than $\log(v)+1$, all $v > v_0$ fulfill the conditions, where $v_0$ is the solution of $v_0 \log(2) = \log(v_0) + 1$ with $v_0 > 1$ (a numeric approximation is $v_0 \approx 3.053$). To assure the first condition is also fulfilled for finite $d$, we have to add a small factor $w$ which depends on $v$ and $d$. However, for a fixed choice of $v$ it is upper bounded, so that we may choose $m = vd + w$.

For the choice $v, w = 4$, i.e., $m = 4(d+1)$ we obtain a first derivative $(4^{2+d}(x/(e+ed))^x(1+(1+d)(5+4d)\log((4d)/(e+ed))))/((1+d)(5+4d)^2)$ which for the sake of setting $= 0$ simplifies to $1+(1+d)(5+4d)\log((4d)/(e+ed))$. This function is monotonously growing for $d \geq 0$, negative at $d = 1$ and positive at $d = 2$. Thus, checking $2^{4(d+1)}/((e4(d+1)/d)^d(4(d+1)+1)) > 1$ suffices for $d = 1,2$, which holds true, indeed.

**Mismatch of the Second Kind:** Let $S_1, S_2 \subset \mathcal{X}$ be disjoint sets that are shattered by $\mathcal{H}$ but $S := S_1 \cup S_2$ is not shattered by $\mathcal{H}$, thus there exists a labeling $f : S \to \{0,1\}$ with $f \notin \mathcal{H}_{|S}$ but $f_{|S_i} \in \mathcal{H}_{|S_i}$. Define $Y = f(X)$ and put equal weight on those points and only those points in $S_1$ during the first half and analogous for $S_2$ and the second half. Then there is virtual drift only and also model drift. To see that $S_1, S_2$ exist, first choose $S_1 \subset \mathcal{X}$ with size $d$, which exists because $\mathcal{H}$ has VC-dimension $d$. Because $|\mathcal{H}| > 2^{|S_1|}$, there has to exist at least one point $x \in \mathcal{X} \setminus S_1$ such that $2^1 \geq |\mathcal{H}_{|\{x\}}| > 1$. Thus, the set $S_2 = \{x\}$ is shattered by $\mathcal{H}$, $S_1$ and $S_2$ are disjoint, $S = S_1 \cup S_2$ has size $d+1$ and thus cannot be shattered by $\mathcal{H}$. $\square$

The proof is based on the idea that, since the model complexity is bounded, the model either cannot distinguish enough points to notice the drift or cannot match the entire posterior. Notice that the proof heavily relies on the assumption of a finite VC-dimension to construct the drift process. A possible justification for this assumption is that it is necessary to control the discrepancy between the detection equation and its empirical counterpart as is needed for statistical tests. An obvious question is whether the statement stays true if we drop this assumption. However, similar findings actually apply under mild assumptions and thus in many real-world scenarios: As shown in Theorem 1, supervised drift detection is equivalent to conditional independence tests, a testing problem which in stark contrast to (unconditional) independence is known to not admit a non-trivial test with statistical power (Shah and Peters, 2020).

Reconsidering the loss-based drift detectors, we can predict four different outcomes: If the loss does not change either (1a) there is no real drift, or (1b) the models are not complex enough, e.g., because they are over-regularized, to match the structure and smooth out the drift. If the loss does change either (2a) there is real drift, or (2b) the models are not complex enough to match the entire structure at once. Notice that in both problematic cases (b) there is an issue with the model's complexity. This is not surprising, as the proof of Theorem 3 is based exactly on this idea. This idea extends to arbitrary learning algorithms as those have to make a trade-off between model complexity and convergence which results in case (b), too.

This shows that the usually assumed connection between real drift and change of model accuracy is not valid for any learning model if we do not make any assumption on the distribution. Thus, for supervised drift detection there does not exist a universal supervised drift detector, in particular, not one that is based on model loss. However, this is of course not true if we restrict the set of all "allowed drift processes". For example: For Gaussian distributions, conditional independence can be tested. Thus, we suggest selecting a model class that is rich enough to learn all expected distributions, but small enough to assure fast convergence.

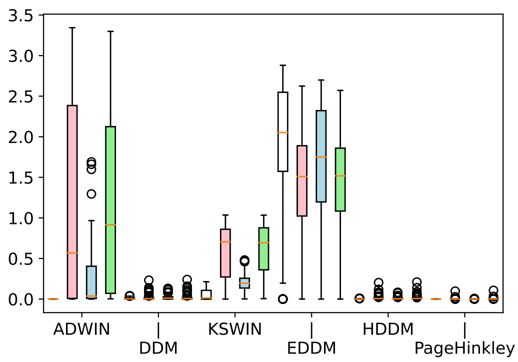
## 4 EMPIRICAL EVALUATION

In the following, we demonstrate our theoretical insights in experimental setups and have a look at the strength of these effects. In particular, we will show that the mismatches actually occur in practical setups and constitute a major concern in stream learning. To do so, we conduct three different experiments: In the first and the second experiment, the ability to detect concept drift is evaluated; in the first experiment, a simplified setup is used to accurately analyze the detection behavior while in the second experiment, a more realistic streaming setup is used where only the timing of the drift is controlled. In the third experiment, we evaluate the effect of false positive detection on the performance of stream learners.

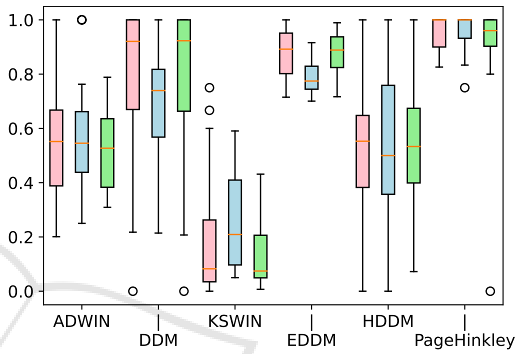
Notice that the methods considered here mainly originate from a stream learning setup which is designed to adapt to the drift, commonly measured by interleaved test-train error, rather than to perform the precise test.

All experiments are performed on the following standard synthetic benchmark datasets AGRAWAL (Agrawal et al., 1993), MIXED (Gama et al., 2004), RandomRBF (Montiel et al., 2018), RandomTree (Montiel et al., 2018), SEA (Street and Kim, 2001), Sine (Gama et al., 2004), STAGGER (Gama et al., 2004) and the following real-world benchmark datasets "Electricity market prices" (Elec; Harries et al., 1999), "Forest Covertype" (Forest; Blackard et al., 1998), and "Nebraska Weather" (Weather; Elwell and Polikar, 2011). To remove uncontrolled effects caused by unknown drift in the real-world datasets, we apply a permutation scheme (Hinder et al., 2022) and induce real drift by a label switch. As a result, all datasets have controlled real drift and no virtual drift. We induce virtual drift by segmenting the data space using random decision trees.

For comparability, all problems are turned into binary classification tasks. Class imbalance is controlled, so it does not exceed 25%/75% in ratio. This way we




(a) Number of alarms




(b) False alarm rate

Figure 1: Evaluation of drift detection. In group order: no-drift (plot (a) only; white), real drift (pink), virtual drift (blue), real and virtual drift (green). Mean taken over dataset and method before turned into a boxplot.



obtained $2 \times 2$ distributions with controlled drifting behavior which we will refer to with the numbers 00 – 11, i.e., $\mathcal{D}_{ij}(X,Y) = \mathcal{D}_i(X)\mathcal{D}_j(Y \mid X)$.

**Drift Detection in a Controlled Setup.** In the first experiment, we analyze the effect of the drift type, i.e., none, real, virtual, and both, on the number of detected drifts and the true positive rate. From each distribution we draw two independent samples $S^{\text{tr}}_{ij}, S^{\text{te}}_{ij} \sim \mathcal{D}_{ij}$ of size 200 for training and testing. We train one model on each training set $S^{\text{tr}}_{ij}$ and evaluate it on a stream which is created by concatenating the test set $S^{\text{te}}_{ij}$ with another test set $S^{\text{te}}_{kl}$. If $i \neq k$ there is virtual drift, if $j \neq l$ there is real drift.

We consider the following models: Decision Tree, Random Forest, $k$-Nearest Neighbour, Bagging (with Decision Tree), AdaBoost (with Decision Tree), Gaussian Naïve Bayes, Perceptron, and linear SVM (Pedregosa et al., 2011). The models are not modified, i.e., no passive adaption, during the evaluation phase.

By evaluating the model on the stream we obtain a stream of losses, i.e., $s_i = \mathbf{1}[h(x_i) \neq y_i]$, to which we apply the drift detector and document the detected

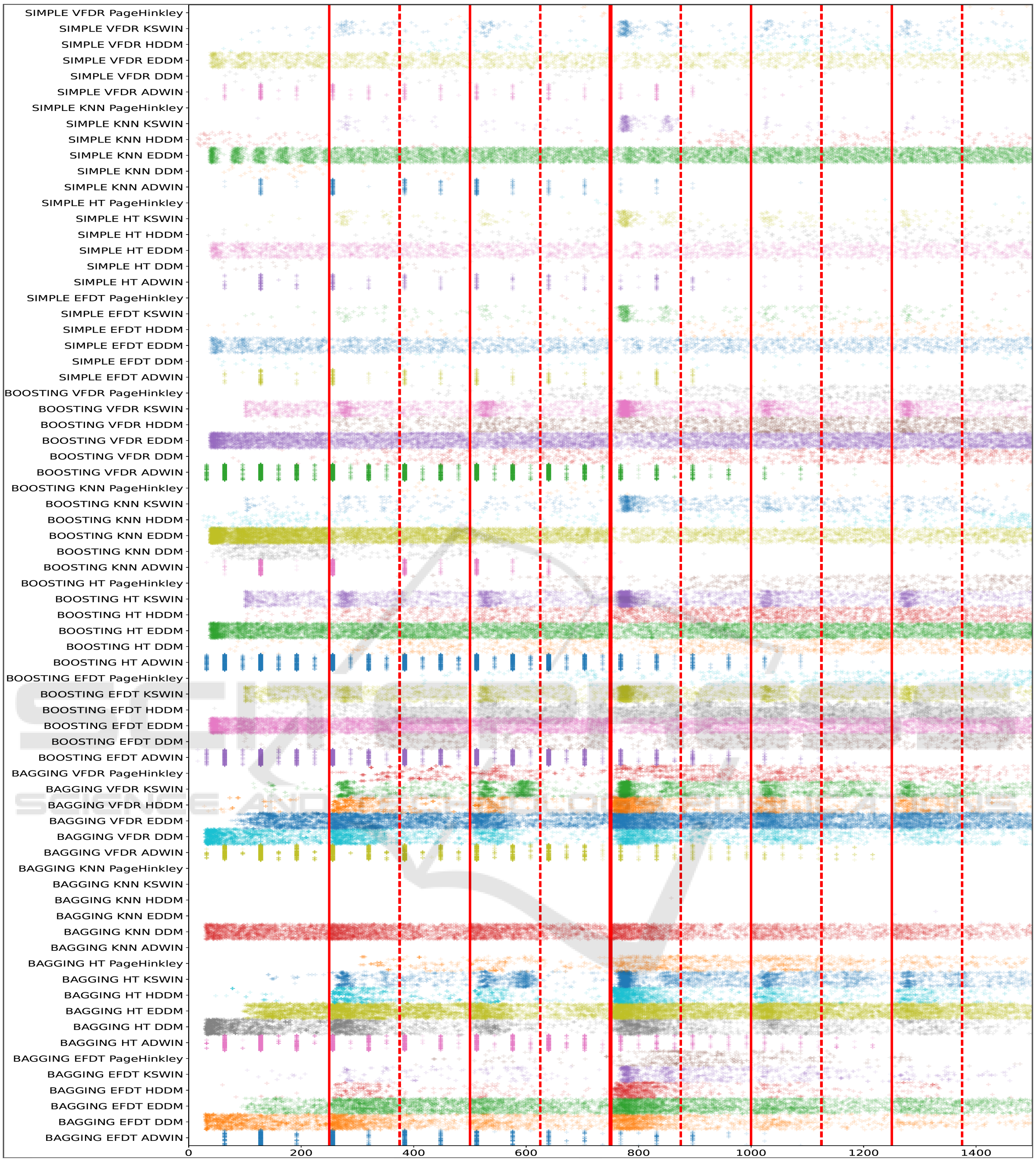



Figure 2: Evaluation of drift detection on Weather based data streams for different model/detector setups. Red lines mark change points (thin: virtual drift, thick: real drift), and dashed lines mark the end of considered windows. Points mark found drifts, $x$-axis shows timepoint/sample number in the stream of detection, $y$-axis shows setup and run.

Table 1: Mean F1-score (200 runs) of real drift detection.

| Method | | | Elec | Forest | RBF | SEA | STAGGER | Weather |
|---|---|---|---|---|---|---|---|---|
| BAGGING | HT | HDDM | 0.49±0.19 | 0.23±0.28 | 0.47±0.36 | 0.00±0.07 | 0.99± 0.09 | 0.48±0.21 |
| | KNN | DDM | 0.18±0.04 | 0.17±0.07 | 0.19±0.05 | 0.19±0.08 | 0.74± 0.22 | 0.18±0.05 |
| BOOSTING | EFDT | KSWIN | 0.24±0.06 | 0.15±0.14 | 0.28±0.12 | 0.10±0.22 | 0.81± 0.25 | 0.25±0.06 |
| | HT | ADWIN | 0.11±0.13 | 0.03±0.10 | 0.16±0.12 | 0.01±0.05 | 0.15± 0.18 | 0.15±0.12 |
| | | KSWIN | 0.23±0.05 | 0.20±0.16 | 0.26±0.10 | 0.11±0.25 | 0.86± 0.20 | 0.23±0.05 |
| | VFDR | KSWIN | 0.26±0.06 | 0.21±0.18 | 0.27±0.10 | 0.10±0.22 | 0.83± 0.20 | 0.24±0.05 |
| SIMPLE | VFDR | ADWIN | 0.17±0.25 | 0.01±0.10 | 0.11±0.23 | 0.00±0.00 | 0.52± 0.16 | 0.16±0.25 |
| | | KSWIN | 0.73±0.33 | 0.13±0.30 | 0.43±0.46 | 0.00±0.07 | 0.99± 0.10 | 0.52±0.36 |
| ShapeDD (unsupervised) | | | 0.36±0.25 | 0.29±0.23 | 0.45±0.14 | 0.35±0.08 | 0.49± 0.14 | 0.41±0.20 |

Table 2: Mean number of alerts in stream blocks (200 runs). In block ordering is non-drifting (max. 7), real drift (max. 1), virtual drift (max. 4).

| Method | | | Elec | Forest | RBF | SEA | STAGGER | Weather |
|---|---|---|---|---|---|---|---|---|
| BAGGING | HT | HDDM | 1.03 0.98 1.46 | 1.06 0.48 1.58 | 0.56 0.72 0.90 | 0.01 0.00 0.01 | 0.02 1.00 0.00 | 0.88 0.92 1.62 |
| | KNN | DDM | 5.38 0.96 3.58 | 5.34 0.96 3.62 | 5.24 0.95 3.56 | 3.98 0.90 3.32 | 0.44 1.00 0.38 | 5.20 0.99 3.54 |
| BOOSTING | EFDT | KSWIN | 3.90 0.98 2.93 | 3.68 0.64 2.39 | 2.77 0.87 2.43 | 1.12 0.21 0.70 | 0.33 0.96 0.17 | 3.85 1.00 2.98 |
| | HT | ADWIN | 3.91 0.42 1.97 | 2.50 0.11 1.42 | 4.08 0.64 2.06 | 1.74 0.05 1.06 | 2.59 0.50 1.50 | 4.30 0.66 2.13 |
| | | KSWIN | 4.00 1.00 2.92 | 2.69 0.66 2.28 | 3.16 0.98 2.84 | 0.86 0.16 0.70 | 0.31 1.00 0.19 | 3.82 1.00 3.24 |
| | VFDR | KSWIN | 3.53 0.99 2.74 | 2.28 0.64 2.24 | 2.98 0.94 2.70 | 0.88 0.20 0.69 | 0.34 1.00 0.27 | 3.50 0.99 2.96 |
| SIMPLE | VFDR | ADWIN | 1.03 0.32 0.75 | 0.44 0.02 0.34 | 1.06 0.18 0.70 | 0.04 0.00 0.04 | 1.10 0.94 0.78 | 1.25 0.30 0.72 |
| | | KSWIN | 0.27 0.84 0.32 | 0.26 0.18 0.50 | 0.18 0.50 0.44 | 0.00 0.01 0.00 | 0.00 0.99 0.00 | 0.30 0.74 0.90 |
| ShapeDD (unsupervised) | | | 0.21 0.72 2.10 | 0.14 0.66 2.52 | 0.14 1.00 2.65 | 0.04 0.97 3.64 | 0.18 1.00 2.21 | 0.18 0.86 2.34 |

drifts. We consider the following drift detectors: ADWIN (Bifet and Gavaldà, 2007), DDM (Gama et al., 2004), EDDM (Baena-García et al., 2006), HDDM-A (Frías-Blanco et al., 2015), KSWIN (Raab et al., 2019), and PageHinkley (Page, 1954).

We present the total number of found drift events per run in Figure 1a and the rate of false alarms in Figure 1b, where we consider an alarm as a true positive if it was observed during the first 125 samples after the drift. If it was observed before, i.e., when train and test distribution coincide, or with a very large delay, i.e., after more than 125 into the second part of the stream, the detection is considered to be a false positive. We allow multiple true positives per stream. If a stream does not result in a single detection it is excluded from the alarm rate analysis.

As can be seen, most of the methods do not yield any alarms. The ones that do (ADWIN, KSWIN, and EDDM) show strong variation in the false alarm rate, where KSWIN outperforms the others in this regard. Only with respect to the sensitivity in the case of virtual drift, it is outperformed by EDDM due to the low detection rate. Furthermore, virtual drift is harder to detect than real which is harder than both. Additionally, except for KSWIN, all methods show an extremely high false alarm rate.

**Drift Detection in a Streaming Setup.** To evaluate the decision capabilities in a more realistic setup, we proceed as follows: We make use of standard, active stream learning algorithms and document the time when the internal drift detectors initiate a model reset. We consider both single model approaches (SIMPLE), where only a single model is used, and ensemble approaches, including bagging (BAGGING) and boosting (BOOSTING). We make use of the same drift detectors as before and consider the following models: Hoeffding Tree (HT; Bifet et al., 2010), Sliding Window *k*-NN (KNN; Montiel et al., 2018), Extremely Fast Decision Trees (EFDT; Manapragada et al., 2018), and Very Fast Decision Rules (VFDR; Kosina and Gama, 2013). To verify our claim that unsupervised drift detectors can help to identify the change points, we also consider the Shape-based Drift Detector (ShapeDD; Hinder et al., 2021).

We apply those methods to streams which are constructed as follows: Each stream consists of 6 blocks consisting of 250 datapoints, each from a single distribution with drift between consecutive blocks. The distributions are constructed as described above – each stream is constructed based on a single dataset. Except for the change from block 1 to 2 and 3 to 4, all blocks have virtual drift only, from 1 to 2 there is no drift, from 3 to 4 there is virtual and real drift.

We report the mean F1-score (Table 1) and the

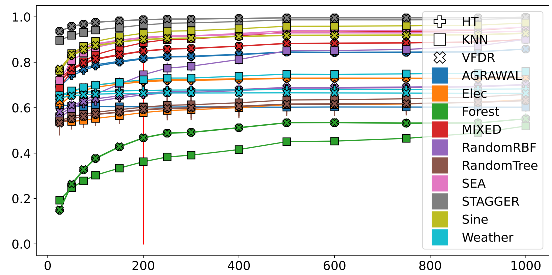



Figure 3: Effect of alarm rate on model accuracy. $x$-axes shows reset frequency in samples between resets, $y$-axis shows mean interleaved test-train accuracy over 200 runs.

mean number of alerts per drift type (Table 2) for a selection of method combinations and ShapeDD. We split each of the 6 blocks into a first and second half and accumulate the detection in each half block. The first half of the 4th block which corresponds to the real drift is considered as a positive, all others are as negatives. Furthermore, we visualized the results on the stream based on the Weather dataset in Figure 2.

Considering Figure 2, it can be seen that many drift detectors produce far too many alarms, which fits the expectation of the first experiment. In some cases, the found drifts seem to be nearly independent of the actual drift events. Only KSWIN shows a solid performance in all setups. Furthermore, the SIMPLE setup produces far fewer drift events, which is to be expected as they make use of a single drift detector only. These results are confirmed on the additional datasets. As one can see in Table 1, for most supervised drift detectors we observe low F1-scores. As can be seen in Table 2, this is caused by high false positive rates. Indeed, as can easily be seen, many methods also detect drift in the non-drifting blocks. In contrast, we observe that the unsupervised drift detection method ShapeDD obtains comparably high F1-scores even though false alarms are expected, i.e., in case of a perfect unsupervised detection the obtain F1-score is 0.33. Reconsidering Table 2, ShapeDD usually shows the lowest number of detection in non-drifting blocks. The only exceptions are SEA, where the other methods with low non-drifting defections do not detect anything, and STAGGER, which appears to be a particularly simple dataset for all stream-learning methods. This perfectly aligns with the theoretical considerations (Corollary 1) and supports our proposal to combine unsupervised drift detection with a suitable statistical test to perform supervised drift detection rather than a loss-based approach.

**Effect of False Alarms on Performance.** To study the effect of many false alarms, we used the same datasets as before to create streams of a length of 1,000 samples without drift. We apply Hoeffding Tree (Bifet et al., 2010), Sliding Window $k$-NN (Montiel et al., 2018), and Very Fast Decision Rules (Kosina and Gama, 2013) to each of the streams, reset the model after every 25, 50, 75, 100, 150, 200, 250, 300, 400, 500, 600, 750, 900, and 1,000 samples, respectively, to simulate different false alarm rates, and document the interleaved test-train error. We repeat the process 200 times. The results are shown in Figure 3. As expected, mean accuracy grows anti-proportionally to the number of resets (Spearman's $\rho$, $p < 0.001$).

Also, observe that after 200 samples nearly all methods reach a plateau which justifies the choice of this sample size in the other experiments.

## 5 CONCLUSION

In this work, we considered the interconnection of concept drift, statistical tests, and learning algorithms from a theoretical point of view. We provided a generalized notion of real drift and showed that, other than in the unsupervised setup, a universal supervised drift detector cannot exist. Considering loss-based supervised drift detection we found that the connection between real drift and learning models is not valid if we do not make assumptions about the distributions. In particular, this approach is not suited for monitoring setups. In our experimental evaluation, we demonstrated that unsupervised drift detection constitutes a good choice for this setting. When applying online learning without additional knowledge guiding the choice of a suitable model, considering drift detection on the joint distribution might be a valuable option. Besides, we found that updating the model very frequently due to false alarms is decreasing the performance of online learners. For practical applications, this indicates, that one should carefully select a suitable model according to the data.

## REFERENCES

Agrawal, R., Imielinski, T., and Swami, A. N. (1993). Database mining: A performance perspective. *IEEE Trans. Knowl. Data Eng.*, 5:914–925.

Baena-García, M., Campo-Ávila, J., Fidalgo-Merino, R., Bifet, A., Gavald, R., and Morales-Bueno, R. (2006). Early drift detection method.

Bifet, A. and Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA*, pages 443–448.

Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B., and Braun,

M. (2010). Moa: Massive online analysis. *Journal of Machine Learning Research 11: 1601-1604*.

Blackard, J. A., Dean, D. J., and Anderson, C. W. (1998). Covertype data set.

Bu, L., Alippi, C., and Zhao, D. (2016). A pdf-free change detection test based on density difference estimation. *IEEE transactions on neural networks and learning systems*, 29(2):324–334.

Ditzler, G., Roveri, M., Alippi, C., and Polikar, R. (2015). Learning in nonstationary environments: A survey. *IEEE Comp. Int. Mag.*, 10(4).

Eliades, D. G. and Polycarpou, M. M. (2010). A Fault Diagnosis and Security Framework for Water Systems. *IEEE Transactions on Control Systems Technology*, 18(6):1254–1265.

Elwell, R. and Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531.

Frías-Blanco, I., d. Campo-Ávila, J., Ramos-Jiménez, G., Morales-Bueno, R., Ortiz-Díaz, A., and Caballero-Mota, Y. (2015). Online and non-parametric drift detection methods based on hoeffding's bounds. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):810–823.

Gama, J., Medas, P., Castillo, G., and Rodrigues, P. P. (2004). Learning with drift detection. In *Advances in Artificial Intelligence - SBIA 2004, 17th Brazilian Symposium on Artificial Intelligence, São Luis, Maranhão, Brazil, September 29 - October 1, 2004, Proceedings*, pages 286–295.

Gama, J. a., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):44:1–44:37.

Gonçalves Jr, P. M., de Carvalho Santos, S. G., Barros, R. S., and Vieira, D. C. (2014). A comparative study on concept drift detectors. *Expert Systems with Applications*, 41(18):8144–8156.

Gözüaçık, Ö., Büyükçakır, A., Bonab, H., and Can, F. (2019). Unsupervised concept drift detection with a discriminative classifier. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2365–2368.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. (2006). A kernel method for the two-sample-problem. volume 19.

Hanneke, S., Kanade, V., and Yang, L. (2015). Learning with a drifting target concept. In *Int. Conf. on Alg. Learn. Theo*, pages 149–164. Springer.

Harries, M., cse tr, U. N., and Wales, N. S. (1999). Splice-2 comparative evaluation: Electricity pricing. Technical report.

Hinder, F., Artelt, A., and Hammer, B. (2019). A probability theoretic approach to drifting data in continuous time domains. *arXiv preprint arXiv:1912.01969*.

Hinder, F., Artelt, A., and Hammer, B. (2020). Towards non-parametric drift detection via dynamic adapting window independence drift detection (dawidd). In *ICML*.

Hinder, F., Brinkrolf, J., Vaquet, V., and Hammer, B. (2021). A shape-based method for concept drift detection and signal denoising. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 01–08. IEEE.

Hinder, F., Vaquet, V., and Hammer, B. (2022). Suitability of different metric choices for concept drift detection. In *International Symposium on Intelligent Data Analysis*, pages 157–170. Springer.

Hu, H., Kantardzic, M., and Sethi, T. S. (2020). No free lunch theorem for concept drift detection in streaming data classification: A review. *WIREs Data Mining and Knowledge Discovery*, 10(2):e1327.

Kosina, P. and Gama, J. (2013). Very fast decision rules for classification in data streams. *Data Mining and Knowledge Discovery*, 29:168–202.

Losing, V., Hammer, B., and Wersing, H. (2015). Interactive online learning for obstacle classification on a mobile robot. In *2015 international joint conference on neural networks (ijcnn)*, pages 1–8. IEEE.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. (2018). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363.

Manapragada, C., Webb, G. I., and Salehi, M. (2018). Extremely fast decision tree. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1953–1962.

Mohri, M. and Muñoz Medina, A. (2012). New analysis and algorithm for learning with drifting distributions. In *Int. Conf. on Alg. Learn. Theo*, pages 124–138. Springer.

Montiel, J., Read, J., Bifet, A., and Abdessalem, T. (2018). Scikit-multiflow: A multi-output streaming framework. *Journal of Machine Learning Research*, 19(72):1–5.

Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1-2):100–115.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Raab, C., Heusinger, M., and Schleif, F.-M. (2019). Reactive soft prototype computing for frequent reoccurring concept drift. In *ESANN*.

Shah, R. D. and Peters, J. (2020). The hardness of conditional independence testing and the generalised covariance measure. *The Annals of Statistics*, 48(3):1514–1538.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Street, W. N. and Kim, Y. (2001). A streaming ensemble algorithm (SEA) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001*, pages 377–382.

Vaquet, V., Menz, P., Seiffert, U., and Hammer, B. (2022). Investigating intensity and transversal drift in hyperspectral imaging data. *Neurocomputing*, 505:68–79.

Webb, G. I., Lee, L. K., Petitjean, F., and Goethals, B. (2017). Understanding concept drift. *CoRR*, abs/1704.00362.