ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

# Self-Supervised Learning from Incrementally Drifting Data Streams

Valerie Vaquet[1,2], Jonas Vaquet[1], Fabian Hinder[1], Kleanthis Malialis[2], Christos G. Panayiotou[2,3], Marios M. Polycarpou[2,3] and Barbara Hammer[1] *

1- Machine Learning Group
Bielefeld University, Bielefeld - Germany

2-KIOS Research and Innovation Center of Excellence,
University of Cyprus, Nicosia - Cyprus

3-Department of Electrical and Computer Engineering,
University of Cyprus, Nicosia - Cyprus

**Abstract**.
Supervised online learning relies on the assumption that ground truth information is available for model updates at each time step. As this is not realistic in every setting, alternatives such as active online learning, or online learning with verification latency have been proposed. In this work, we assume that no label information is available after intitial training. We argue that provided we can characterize the expected concept drift as incremental drift, we can rely on a self-labeling strategy to keep updated models. We derive a $k$-NN-based self-labeling online learner implementing the presented self-supervised scheme and experimentally show that this is an option for learning from incrementally drifting data streams in the absence of label information.

## 1 Introduction

In many scenarios, data is arriving as a non-stationary data stream requiring machine learning algorithms to adapt to the most recent data flexibly. As this setting poses additional challenges to that of batch learning, where the entire dataset is available, considerable research is dedicated to this assignment with the majority considering *supervised online learning* as a setup [1]. Here, the fundamental assumption is that after a model provides its prediction, the ground truth becomes available for updating the model. As this is not realistic in many real-world applications, some contributions focus on verification latency [2] or alternative setups such as active learning [3] or semi-supervised approaches [4].

Mostly, these still assume that some ground truth becomes available. In contrast, we are interested in online learning strategies which, after an initial set-up phase, proceed without any label information. Here so-called *concept drift* constitutes a particular challenge, i.e., if the underlying distribution changes and initial models might become invalid. In this work we focus on the scenario of

*We gratefully acknowledge funding from the European Research Council (ERC) under the ERC Synergy Grant Water-Futures (Grant agreement No. 951424), the European Union's Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE), and the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

incrementally drifting data streams; we leverage self-labeling strategies similar to those that have been proposed for semi-supervised learning in the batch setup [3, 5]. In particular, as an exemplary self-supervised learning strategy, we propose a window-based $k$-NN classifier relying on certainty scores to decide whether its predictions are used for updating the model.

This paper is structured as follows: After formulating the setup of supervised online learning (Section 2.1), we analyze distributional changes in more detail (Section 2.2), which leads to an alternative setting that assumes no ground truth after an initial training phase (Section 2.3). In Section 3, we derive a simple strategy to learn in this setting (Section 3). Finally, we will experimentally evaluate the proposed method (Section 4) and conclude this work (Section 5).

## 2 Online Learning from Drifting Data Streams

We will first define the setup of supervised online learning and concept drift. Afterward, we will investigate how the signals we observe are composed to analyze which components are drifting with which properties. Finally, we will propose the setup of self-supervised online learning.

### 2.1 Supervised Online Learning

Let $\mathcal{X} \subset \mathbb{R}^d$ be the feature, $\mathcal{Y}$ be the target space, and $\mathcal{T}$ the time domain. A data stream $S = ((X_i, Y_i))_{i=1}^N$ is a (potentially infinite) sequence of $\mathcal{X} \times \mathcal{Y}$-valued random variables. We assume that each sample $(X_i, Y_i)$ is observed at some time point $T_i \in \mathcal{T}$ with $T_i \leq T_{i+1}$ which determines their distribution, i.e., $(X_i, Y_i) \sim \mathcal{D}_{T_i}$ where $\mathcal{D}_t$ is a Markov kernel from $\mathcal{T}$ to $\mathcal{X} \times \mathcal{Y}$. *Drift* takes place if $\mathcal{D}_t \neq \mathcal{D}_s$ for $t \neq s$, or equivalently, if $T_i$ and $(X_i, Y_i)$ are not independent [6]. The goal of supervised online learning is to infer an adaptive model $h_t$ approximating the stream-generating distribution process $\mathcal{D}_t$ at each time point $t$. This is usually done in the test-then-train setup, i.e., at time $T_i$ a new sample is arriving. First a prediction $\hat{Y}_i = h_{T_{i-1}}(X_i)$ is obtained, then $h_{T_i}$ is obtained by updating the model with $(X_i, Y_i)$.

### 2.2 Decomposing the Signal

We are interested in learning strategies without label information $Y_i$. For arbitrary, unknown drift, no valid learning strategies can exist. Hence, we first have a closer look at different types of drift. Drift can occur in the quantity of interest, the environmental impacts, or the measurement process. Formally, $X$ is a distortion of the value of interest $\tilde{X}$ induced by a (time-dependent) Markov kernel $M_t$ from $\mathcal{X} \times \mathcal{T}$ to $\mathcal{X}$ which models measurement errors, i.e., $X \sim M_T(\tilde{X})$. A simple example is Gaussian noise $M_t(\tilde{x}) = \mathcal{N}(\tilde{x}, \sigma)$. In many scenarios $\tilde{X}$ is not affected by drift throughout the stream but only $M_t$. As a practical example, for hyperspectral cameras in a laboratory, the hyperspectral sensors are slowly changing the mean, i.e. $M_t(\tilde{x}) = \mathcal{N}(\tilde{x} + \gamma(t), \sigma)$ for some function $\gamma$ depending on the camera only [7]. In many systems observing natural phenomena or user behavior, seasonal patterns can be observed, e.g. water demands smoothly

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

change with the seasons [8]. These types of drift are incremental drift as defined by [1].

A key observation is that, provided the effect of the drift is smaller than the signal strength induced by the model, incremental learning without label information can rely on self-labeling strategies. Here we refer to stream-learning algorithms which are based on fixed-size sliding windows, as these are often used in practice [1]:

**Theorem 1.** *Let $\mathcal{X} = \mathbb{R}^d, \mathcal{T} = \mathbb{R}$. Let $\gamma : \mathcal{T} \to \mathcal{X}$ be additive shift, i.e. $X = \tilde{X} + \gamma(T)$ and assume no other drift is present, i.e. $Y \leftarrow \tilde{X} \to X \leftarrow T$ is faithful. If $\gamma$ is Lipschitz continuous with constant $L$ then for any margin size $M > 0$ the following relation of margins holds assuming the sliding window does not contain observations that are more than $w$ apart, i.e. $|T - T'| \leq w$:*

$$\begin{aligned}&\mathbb{P}[\|\tilde{X} - \tilde{X}'\| \leq M + wL \Rightarrow Y = Y'] \\ &\leq \mathbb{P}[\|X - X'\| \leq M \Rightarrow Y = Y' \mid |T - T'| \leq w]\end{aligned}$$

*Sketch of proof.* Since $X = \tilde{X} + \gamma(T)$, by applying the triangle inequality and Lipschitz continuity we can conclude $\mathbb{P}[\|\tilde{X} - \tilde{X}'\| > M + wL, \|X - X'\| \leq M \mid |T - T'| \leq w] = 0$. The statement follows. □

Notice, that a generalization of this statement where $\gamma$ is induced by general (stochastic) differential equations is possible but beyond the scope of this paper.

### 2.3 Self-Supervised Online Learning

We consider the following self-supervised learning setup: The data stream $S$ is split into an initial development $S_{\text{dev}} = (X_{i_j}, Y_{i_j})_{j=1}^{N_{\text{dev}}}$ and deployment $S_{\text{dep}} = (X_{i_j})_{j=N_{\text{dep}}}^{N}$ part. During the development phase, a model is trained using the usual test-then-train scheme, while at the deployment phase no ground truth is available for updating the model anymore.

Theorem 1 implies that, if the classification model has a sufficiently large hypothesis margin for a new sample within a data stream where the drift is limited, the label induced by the model is correct with high probability. Based on this fact, we propose to combine a window-based online learning scheme and a self-labeling strategy which infers the label from the given model and adapts the model accordingly. As a proxy for the hypothesis margin, we propose to rely on model-specific certainty estimates of the prediction, as these relate to the epistemic uncertainty of the model prediction, hence the margin [9]. In the next section, we will derive a $k$-NN-based online learning algorithm together with different certainty estimates. We chose a $k$-NN architecture because they are particularly suitable when considering soft margins and have been successfully applied for online learning tasks [10].

## 3 A Self-Teaching Online $k$-NN

We consider an online $k$-NN classifier with a fixed-size sliding window. At the development phase, we train the model according to the test-then-train scheme,

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

adding data as suitable. In the deployment phase, we rely on the model's predictions. As a baseline, the naive version simply updates the model with $(X_i, h_{T_{i-1}}(X_i))$ at each sample, disregarding the hypothesis margin. For a more informed version, we use estimates of the classifier's confidence in its prediction [11]. We consider the following certainty scores:

$$c_1(x, \hat{y}) = \frac{1}{k} \sum_{(x',y') \in N_k(x)} \frac{\tilde{\delta}_{y',\hat{y}}}{1 + d(x, x')} \quad (1) \qquad c_2(x, \hat{y}) = \frac{\sum_{(x',y') \in N_k(x)} \frac{\delta_{y',\hat{y}}}{d(x,x')}}{\sum_{(x',y') \in N_k(x)} \frac{1}{d(x,x')}}, \quad (2)$$

where $d$ is a distance, $\delta_{y,y'}$ is the Kronecker delta, $\tilde{\delta}_{y,y'} := 1 - 2\delta_{y,y'}$, and $N_k(x)$ is the $k$-neighborhood of $x$. Besides, we examine a version which explicitly relates to the margin by referring to the $k$ closes samples with the same $(N_k^+(x, \hat{y}))$/different class $(N_k^-(x, \hat{y}))$ compared to $\hat{y}$

$$c_3(x, \hat{y}) = \frac{1}{k} \sum_{(x',y') \in N_k^+(x)} \frac{1}{1 + d(x, x')} - \frac{1}{k} \sum_{(x',y') \in N_k^-(x)} \frac{1}{1 + d(x, x')} \quad (3)$$

While $c_1$ and $c_2$ only focus on the $k$-neighborhood of $x$, $c_3$ considers the closest $k$ samples of the predicted and any other class thereby including some kind of information on the margin between the samples in the neighborhood. In the deployment phase, the model is only updated by $(x, \hat{y})$ if $c(x, \hat{y}) > \theta$, where $\theta$ is a hyper-parameter of the method.

## 4 Experiments

*Datasets.* We consider two datasets. The *Spectra* dataset contains hyperspectral measurements of sugar and coffee which were collected under laboratory conditions. We add incremental drift to the dataset by applying increasing intensity shifts to the data. We model the drift realistically according to an analysis of hyperspectral data [7]. In total, we consider 56 data streams consisting of 5,000 samples with 288 spectral bands as features.

Besides, we consider a dataset of pressure measurements from the L-Town [8] water distribution network containing leakages or not (*Leaks*). Changing water demands over the seasons inflict incremental drift in the environmental variable. We simulate leakages in 764 positions. For each, we generate a stream where we take the mean daily pressure at 29 sensors and randomly decide whether we take examples from the leaky or non-leaky scenario for each day, resulting in 764 streams containing 365 samples each.

*Setup* We report the accuracies on $S_{\text{dev}}$ for the proposed variants described above and the following baselines: no update during deployment (lower baseline), supervised online learning, and perfect policy, i.e. only update with correctly classified samples (upper baselines). Next to considering the naive version where each prediction is used for updating, for the certainty-based methods, we report the automatically estimated thresholds (optimal split found by decision

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

**Table 1:** Accuracy on $S_{dep}$ (mean±std. deviation) of the experiments on all data streams

| Method | Spectra | Leaks |
|---|---|---|
| No Learn | $0.6431 \pm 0.0223$ | $0.5751 \pm 0.0275$ |
| Supervised | $0.9356 \pm 0.0057$ | $0.9397 \pm 0.0498$ |
| Perfect Policy | $0.9203 \pm 0.0064$ | $0.7425 \pm 0.1626$ |
| Naive | $0.8874 \pm 0.0627$ | $0.6295 \pm 0.1463$ |
| $c_1$ - auto | $0.8973 \pm 0.0454$ | $0.6325 \pm 0.1479$ |
| $c_2$ - auto | $0.8865 \pm 0.0730$ | $0.6183 \pm 0.1371$ |
| $c_3$ - auto | $0.9076 \pm 0.0294$ | $0.6141 \pm 0.1355$ |
| $c_1$ - opt | $0.9169 \pm 0.0069$ | $0.6870 \pm 0.1470$ |
| $c_2$ - opt | $0.9174 \pm 0.0066$ | $0.6707 \pm 0.1469$ |
| $c_3$ - opt | $0.9174 \pm 0.0065$ | $0.6679 \pm 0.1421$ |

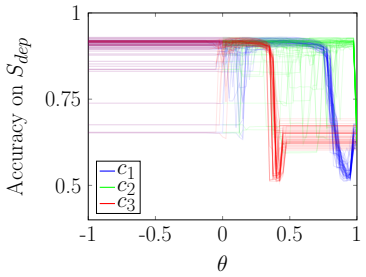


**Fig. 1:** Performance on Spectra for different $\theta$

tree of depth 1) and the results of the optimal threshold for each stream. For the latter we consider the following thresholds $\theta_{c_1}, \theta_{c_3} \in \{-1, -0.975, -0.95, \ldots, 1\}$, $\theta_{c2} \in \{0, 0.025, 0.05, \ldots, 1\}$ for each stream and report the best result, i.e. this is an upper baseline assuming a mechanism to find the most suitable threshold was available. All experiments use Euclidean distance and $k = 5$[1].

*Results* The results are summarized in Table 1. As to be expected, for both datasets we obtain the worst accuracies if we perform no updates during $S_{\text{dep}}$ indicating that the data streams contain concept drift making updates mandatory to keep performance. While for the Spectra dataset updating with a perfect policy almost performs as well as supervised learning, for the Leaks dataset, we obtain much worse results which has to be expected as the dataset contains more noise due to different patterns across weekdays and weekends.

On the Spectra dataset, we observe that choosing a threshold automatically increases the accuracy over the naive updating strategy. However, an improvement can be gained if the optimal threshold is chosen indicating that more work on decision strategies would be beneficial. In converse, for Leaks, we observe that the naive version outperforms some of the proposed scores which is probably caused by the fact that some leakages are difficult to differentiate from the leak-free setting because they only have a limited impact on the pressure measurements due to their location in the system yielding a small margin and potential for error. Again we observe that choosing the optimal threshold benefits the scores considerably although it does not reach the level of the perfect policy results reflecting again that the dataset is more noisy.

While investigating the choice of optimal $\theta$ on the Leaks data streams resulted in very noisy results which is plausible as each stream considers a different leakage location resulting in different network dynamics, analyzing which $\theta$ is optimal for the Spectra streams yielded a more consistent picture. As visualized in Fig. 1, while the optimal threshold for $c_2$ varies a lot, $c_1$ and $c_3$ are yielding much more robust optimal $\theta$. We find that $c1$ and $c_3$ are particularly robust over the considered streams.

[1]The experimental code be accessed at `https://github.com/vvaquet/Self-Supervised-Online-Learning`

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

## 5 Conclusions and future research directions

Since the availability of label information at deployment time is unrealistic in some streaming settings, in this work, we proposed a self-supervised online learning for incrementally drifting data streams. We showcased the suitability of the proposed setup and strategy by evaluating an online $k$-NN implementing the strategy on two types of incrementally drifting data streams.

This work can be understood as a step towards learning from drifting data streams in the absence of ground truth availability. Future work in this setting is required. In particular, a generalization of Theorem 1 where $\gamma$ is induced by general (stochastic) differential equations, is possible, and extremely important for ML as it allows physics-informed ML. Besides, on a more practical note, transferring other online learning mechanisms to this setup, developing and analyzing alternative certainty scores and decision mechanisms is highly relevant. Besides, it might be beneficial to implement a mechanism that can cope with unexpected drift, e.g. by considering drift detection schemes that trigger a recalibration procedure. Finally, both self-supervised and supervised online learning would strongly benefit from novel data benchmarks in which the properties of realistic drifts are documented.

## References

[1] João Gama et al. A survey on concept drift adaptation. *ACM COMPUT SURV*, 46(4):44:1–44:37, March 2014.

[2] Gary R. Marrs, Ray J. Hickey, and Michaela M. Black. The impact of latency on online classification learning with concept drift. In *Knowledge Science, Engineering and Management*, pages 459–469, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[3] Indre Zliobaite, Albert Bifet, Bernhard Pfahringer, and Geoffrey Holmes. Active Learning With Drifting Streaming Data. *IEEE T NEUR NET LEAR*, 25(1):27–39, January 2014.

[4] Heitor Murilo Gomes et al. A Survey on Semi-supervised Learning for Delayed Partially Labelled Data Streams. *ACM COMPUT SURV*, 55(4):1–42, April 2023.

[5] Isaac Triguero, Salvador García, and Francisco Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems*, 42(2):245–284, February 2015.

[6] Fabian Hinder, Valerie Vaquet, and Barbara Hammer. One or two things we know about concept drift—a survey on monitoring in evolving environments. Part A: detecting concept drift. *Frontiers in Artificial Intelligence*, 7:1330257, June 2024.

[7] Valerie Vaquet, Patrick Menz, Udo Seiffert, and Barbara Hammer. Investigating intensity and transversal drift in hyperspectral imaging data. *Neurocomputing*, 505:68–79, 2022.

[8] Stelios G. Vrachimis et al. Battle of the Leakage Detection and Isolation Methods. *J WATER RES PLAN MAN*, 148(12):04022068, December 2022.

[9] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, Mar 2021.

[10] Viktor Losing, Barbara Hammer, and Heiko Wersing. Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing*, 275:1261–1274, 2018.

[11] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.