# Experimental analysis on dissimilarity metrics and sudden concept drift detection

Gerardo Rubino, Sebastián Basterrech, Jan Platoš, Michal Woźniak

# Experimental analysis on dissimilarity metrics and sudden concept drift detection

Sebastián Basterrech, Jan Platoš, Gerardo Rubino and Michał Woźniak

**Abstract** Learning from non-stationary data presents several new challenges. Among them, a significant problem comes from the sudden changes in the incoming data distributions, the so-called concept drift. Several concept drift detection methods exist, generally based on distances between distributions, either arbitrarily selected or context-dependent. This paper presents a straightforward approach for detecting concept drift based on a weighted dissimilarity metric over posterior probabilities. We also evaluate the performance of three well-known dissimilarity metrics when used by the proposed approach. Experimental evaluation has been done over ten datasets with injected sudden drifts in a binary classification context. Our results first suggest choosing the Kullback-Leibler divergence, and second, they show that our drift detection procedure based on dissimilarity measures is pretty efficient.

## 1.1 Introduction

In many real-world problems, a data stream may suddenly change its distributions. This phenomenon is commonly called *concept drift*. One of the most commonly used approaches, which is a reaction to the occurrence of this phenomenon, is its detection with the use of so-called *drift detectors*.

A drift detector signals that the change in data distribution is significant and requires reconstruction or upgrade of the used model [1]. So far, a number of methods have been proposed on how to construct drift detectors. However, most of them require either access to labels or access to prediction metrics of the used prediction model to make a decision. [2]. Other concept drift detectors are based on distances over the underlying data distributions [3–6]. A recent experimental framework for the drift detection evaluation can be found in [7]. To assess a concept drift detector's performance, among the metrics measuring how different two distributions are, some usually considered ones are the number of true positive drift detections, the number of false alarms, the drift detection delay, the confusion matrix, and so on. One difficulty here is that there is typically a cost-benefit trade-off to find between different metrics [8]. The mentioned metrics are more often arbitrarily selected, or they are selected according to the characteristics of the data and the specific problem at hand. Nevertheless, the

metric choice for identifying changes in the probability distributions is a crucial decision addressed in this article. The contributions of this brief paper are two-fold.

(*i*) First, we specify a universal drift detector method in a supervised context without considering any assumptions about data, and we explore the impact of the two main parameters of the proposed technique.

(*ii*) Second, we empirically analyze the performance of our drift detector using three different and important dissimilarity metrics: KL-divergence, Hellinger distance and Wasserstein distance. The selection of these metrics is based on the fact that nowadays are often used in the learning area [5, 9, 10].

Our experimental results over ten simulated datasets with injected drifts show remarkable differences between Hellinger distance and the other two evaluated metrics. In addition, it seems to be also a difference between KL-divergence and Wasserstein distance that makes us provide insights about the advantages of relative entropy in cases of sudden drifts in binary multidimensional data.

This paper also briefly describes the studied dissimilarity metrics. Then, we present the drift detector method and our general methodology. We report the results in Section 1.4. Finally, we conclude with some discussion on further studies.

## 1.2 Background

### 1.2.1 Drift detection problem

Streaming data processing is usually related to problems where data comes in regular data chunks (blocks). Because we focus on the supervised context, we receive a long sequence of (input, output) values organized in chunks of common size $K$.

Consider a system producing the output $\mathbf{y} \in \mathbb{Y}$ when the input is $\mathbf{u} \in \mathbb{U}$. Formally, we receive a time series $\left(C_1, C_2, \ldots\right)$ where $C_i$ is the chunk $i$, composed of the $K$-length sequence $\left(\mathbf{z}_i^{(1)}, \ldots, \mathbf{z}_i^{(K)}\right)$, with $\mathbf{z}_i^{(k)} = \left(\mathbf{u}_i^{(k)}, \mathbf{y}_i^{(k)}\right)$ (that is, $\mathbf{z}_i^{(k)}$ is the (input,output) pair of the $k$th element in the $i$th chunk). We see the elements $\left(\mathbf{z}_i^{(1)}, \ldots, \mathbf{z}_i^{(K)}\right)$ as a sample having size $K$ of a random variable $\mathbf{z}$ over a discrete set $\mathbb{U} \times \mathbb{Y}$. The concept drift idea refers to the phenomenon that the probability distribution of $\mathbf{z}$ changes over time, i.e. there exists a point $t$ such that the underlying distribution of $\{\ldots, \mathbf{z}_{t-2}, \mathbf{z}_{t-1}, \mathbf{z}_t\}$ is different from the distribution of $\{\mathbf{z}_{t+\Delta}, \mathbf{z}_{t+\Delta+1}, \ldots\}$. We refer to *sudden* (known also as *abrupt*) drift when $\Delta = 1$ [3, 11]. Observe that the change in the joint distribution can be provoked either by a change in the posterior distribution ($\Pr(\mathbf{y} \mid \mathbf{u})$) (referred as the *real concept drift*) or by a change in the independent variables collected in $\mathbf{u}$ (referred as the *virtual concept drift*) [11]. In this contribution, we are focusing only on the studies of abrupt real drifts.

### 1.2.2 Re-visiting the concepts on dissimilarities

Let us consider discrete probability distributions, our case of interest. The context is the following: we have two discrete probability distributions (two probability mass functions, pmfs) $p$ and $q$, defined on some common space $\mathbb{S}$, and we want to measure how different they are. We review three different ways proposed for this purpose in computer science applications, primarily used in data mining.

***Kullback-Leibler divergence.***
The Kullback-Leibler (KL) divergence (also abusively called *distance*) between $p$ and $q$ (better said from $p$ to $q$) is [9]

$$\mathrm{KL}(p \parallel q) = \sum_{s \in \mathbb{S}} p(s) \log \frac{p(s)}{q(s)}. \tag{1.1}$$

Observe that this is not strictly a distance as the other dissimilarities analyzed here.It is positive and its value is zero if and only if $p$ and $q$ are identical. In information theory, we know that $\mathrm{KL}(p \parallel q)$ is the quantity of information

lost when we use $q$ instead of $p$, or as an approximation of $p$. The KL divergence does not satisfy the triangular inequality, in general. Several variations of the canonical KL divergence have been introduced in the literature to reach the symmetry property. Also, observe that the expression defining this divergence needs the sum taken for all values $s$ where $p$ and $q$ are not zero. This leads to some technical issues relevant to our work. In case of zero values, a correction is proposed, see [4, 9]. Despite the mentioned inconveniences, the KL divergence also has several advantages, such that: there exists a relationship with the expected value of likelihood ratio, several hypothesis tests are equivalent to KL divergence, and in case of some specific distributions KL divergence computation can be performed very fast Pinsker's inequality, and so on. For more details, please see [4, 9, 12].

***Hellinger distance.*** The definition is as follows [13]:

$$\mathrm{H}(p,q) = \left( \sum_{s\in\mathbb{S}} \big(\sqrt{p(s)} - \sqrt{q(s)}\,\big)^2 \right)^{1/2}. \quad (1.2)$$

This is a distance, so it is equal to zero if both distributions are the same [5]. A particularly interesting property of the Hellinger distance is that it is bounded, the $\mathrm{H}(p,q)$ values are in $[0,\sqrt{2}]$.

***Wasserstein distance.*** Let $\Gamma = \Gamma(p,q)$ stands for the set of pmfs on $\mathbb{S}^2$ having $p$ and $q$ as marginals. Then, given some real $\nu \geq 1$, the $\nu$-Wasserstein distance $\mathrm{W}_\nu(p,q)$ between the two distributions is

$$\mathrm{W}_\nu(p,q) = \inf_{f\in\Gamma} \left( \mathrm{Exp}_f \left( \big[\mathrm{dist}(X,Y)\big]^\nu \right) \right)^{1/\nu},$$

where dist(.,.) denotes the Euclidean distance and $(X,Y)$ is a pair of random variables having distribution $f \in \Gamma$.

The implementation of this distance has technical issues [10], and the usual approach is to get approximations of the theoretical value. This is provided by available packages, like the one used in this paper (see below).

## 1.3 Methodology

**Computation of dissimilarity scores.** For computing the dissimilarities between two distributions, first we need to build a *descriptor* of the distribution of the data [3]. Here, we use the standard estimator as a descriptor based on the binning strategy. Following the previous notation, we receive in the $i$th chunk a $K$-length sequence $\big(\mathbf{z}_i^{(1)},\ldots,\mathbf{z}_i^{(K)}\big)$, with $\mathbf{z}_i^{(k)} = \big(\mathbf{u}_i^{(k)},\mathbf{y}_i^{(k)}\big)$. Then, for the latter, that is, for the output values of the system, the number $\Pr(\mathbf{y}_i = \ell)$, for any $\ell \in \mathbb{Y}$, is naturally estimated by its standard estimator

$$\widetilde{\Pr}(\mathbf{y}_i = \ell) = \frac{1}{K}\sum_{k=1}^{K} 1(\mathbf{y}_i^{(k)} = \ell),$$

where we denote by $1(P)$ the indicator function.For an input $\mathbf{u}$ to the system, we apply the binning strategy decomposing the input space $\mathbb{U}$ into $J$ disjoint "bins" $b^{(1)}, b^{(2)}, \ldots, b^{(J)}$. For the the $C_i$th chunk the conditional probability of having a class $\ell$ in a specific bin $b_i^j$ is estimated by

$$\widetilde{\Pr}\big(\mathbf{y}_i = \ell \,|\, \{\mathbf{u}_i : \mathbf{u}_i^k \in b_i^j, \forall k\}\big) = \frac{\sum_{k=1}^{K} 1((\mathbf{u}_i^k \in b_i^j) \cap (\mathbf{y}_i^k = l))}{\sum_{k=1}^{K} 1(\mathbf{u}_i^k \in b_i^j)}. \quad (1.3)$$

Then, by applying expression (1.3) for each output class $\ell$ we may compute the probability mass function $\widetilde{\Pr}(\mathbf{y}_i \,|\, \{\mathbf{u}_i : \mathbf{u}_i^k \in b_i^j, \forall k\})$. Hence, each bin has associated a pmf, and then we evaluate a change through any two chunks $C_i$ and $C_t$ as follows

$$d_{i,t}^j = \phi\Big(\widetilde{\Pr}(\mathbf{y}_i \,|\, \{\mathbf{u}_i : \mathbf{u}_i^k \in b_i^j, \forall k\}), \widetilde{\Pr}(\mathbf{y}_t \,|\, \{\mathbf{u}_t : \mathbf{u}_t^k \in b_t^j, \forall k\})\Big), \quad (1.4)$$

where $\phi(\cdot)$ is any selected function for estimating the distribution dissimilarity. Finally, we aggregate the estimated dissimilarity for covering the whole input space (for all the bins)

$$\Phi(C_i, C_t) = \frac{1}{J}\sum_{j=1}^{J} d_{i,t}^{j}. \quad (1.5)$$

Finally, we modify the previous aggregation form using a weighted sum. We consider the chance of sampling in a specific bin

$$\Phi(C_i, C_t) = \frac{1}{J}\sum_{j=1}^{J} \gamma_i^j d_{i,t}^{j}, \quad (1.6)$$

where the weight $\gamma_i^j$ is the probability estimation of sampling in a specific region $b_i^j$ of the reference chunk $S_i$

$$\gamma_i^j = \frac{1}{K}\sum_{k=1}^{K} 1(\mathbf{u}_i^k \in b_i^j).$$

***Decision rule using a variance-based threshold.*** Now, let us consider windows of chunks $W_1, W_2, \ldots$ where $W_i = \big(C_i, C_{i+1}, \ldots, C_{i+N-1}\big)$. We employ the previous approach again to look for changes in the data distributions but see a block of chunks as a sliding window on the series of chunks, having $KN$ instances. We proceed as before, except that instead of comparing two windows starting at chunks $C_i$ and $C_{i+1}$, we shift the blocks by $N$ individual chunks, which is, we compare the window starting at chunk $i$ with the one starting at chunk $i + N$. For each $N$ individual chunk is possible to compute a new dissimilarity score by collecting the chunks in batches (windows) and computing a dissimilarity score $\Phi(W_j, W_{j+1})$ applying expression (1.6). Therefore, a sequence of dissimilarity scores is generated $\Phi(W_1, W_2)$, $\Phi(W_2, W_3)$, $\ldots$, $\Phi(W_j, W_{j+1})$. It is necessary to define a procedure for identifying locations where critical points occur to make an automatic decision. Let $m_j$ be the mean of the dissimilarity scores until the last processed window $W_j$, and $\sigma_k$ the standard deviation of this sequence. Given a new dissimilarity score value $\Phi(W_j, W_{j+1})$, we decide that a drift occurs when

$$\Phi(W_j, W_{j+1}) \notin [m_k - \alpha\sigma_k, m_k + \alpha\sigma_k], \quad (1.7)$$

where $\alpha$ is a threshold parameter. This specific decision rule is inspired by techniques for artifact and outliers detection [14, 15]. Window length and $\alpha$ value are the main parameters of the method. A larger $\alpha$ may increase the chance of false negatives. When $\alpha$ is too small, then the chances of false positives increase. Here, we analyze only scenarios where the windows are disjoint, and the $\alpha$ values are static (we don't modify them according to changes in the data). Another parameter that has an impact on the results is the number of bins. It impacts the pmf estimation. A large number also increases computational costs. After a preliminary evaluation, we decided to present results using $J = 5 \times dim(\mathbb{U})$ homogeneous bins, where $dim(\mathbb{U})$ denotes the dimensionality of the input space.

***Methodological approach overview.*** The concept drift detector method analyzed here is summarized in the high-level workflow presented in Figure 1.1. It has the following main steps:

- $(i)$ Homogeneous partition of the input space. We decompose the input space into disjoint bins using parameterized range constraints. The search for the best splitting hyperplanes in $\mathbb{U}$ is out of the scope of this paper. Here, we decided to create homogeneous partitions following the standard binning strategy.
- $(ii)$ Posterior probability estimation. The probability mass function is estimated by applying the expression (1.3). Note that the conditional distribution is made for each partition of the input space.
- $(iii)$ Dissimilarity metric aggregation. In this step, we apply a weighted dissimilarity (expression (1.6)) for computing an aggregated score among the values computed in each partition.
- $(iv)$ Decision rule. Given a new batch of data, we identify either a drift occurred or not using expression (1.7).

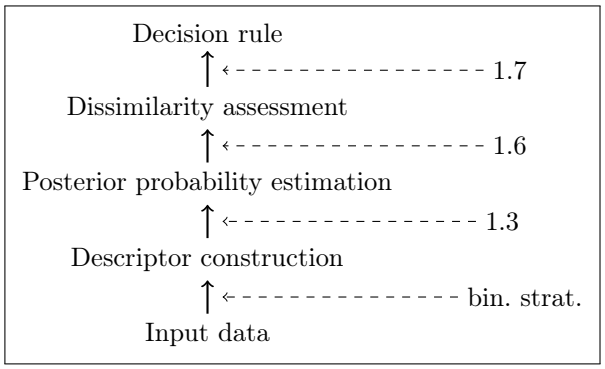


Fig. 1.1: High-level flowchart of investigated algorithm.

## 1.4 Experimental study

In the previous section, we provided a framework for explicitly monitoring the data stream and detecting if a drift occurs. We hypothesize that proposed measures could be used as the base for a decision about drift. We designed experiments to compare the performance of the three mentioned earlier dissimilarity metrics. We used simulated data streams where the drift appearances are marked. In this ongoing work, we study only binary datasets with injected sudden concept drifts. The analyzed window lengths are $\{250, 500, 1000, 2000, 4000\}$, and we studied $\alpha$ values over a large domain (the specific range depended on the metric).

***Benchmark data streams.*** We employed 10 datasets in our performance evaluation studies. We generated 5 binary datasets with 3 features and 5 datasets with 5 features, all of them were created using the *stream-learn* library [16]. Each data stream has 10000 chunks with 250 instances and 20 induced sudden concept drifts. The *stream-learn* library is useful for generating a wide range of datasets with injected drifts, it has the additional advantage that provides the time-stamp where the drifts were injected. The *stream-learn* simulator has a parameter that determines how sudden the change of drift concept is. We used the maximum allowed value for this parameter. More details about the simulation of data streams with sudden drift concepts are specified in [16].

***Performance evaluation.*** We chose the standard metrics: sensitivity, precision, balance accuracy score (BAC) and F1-score [7, 8].

***Results.*** According to our empirical results, we do not appreciate notable differences between results over data with 3 and 5 features. However, we obviously cannot affirm similar behavior in larger input space dimensions. Results obtained by the KL-divergence dissimilarity, the Hellinger distance, and the Wasserstein distance are presented in the figures 1.2, 1.2, and 1.4, respectively. Each of these figures has two graphics, in the left graphic is presented the specificity according to the window length, and the right side is shown the precision according to the window length. We present the results of the specificity metric over datasets with 5 features and the precision obtained over data with 3 features. Each graphic has several curves resulting from different experiments over five datasets. A common behavior in the figures is that the window length is a relevant parameter, which is intuitive because it directly affects the distribution estimation. Another characteristic is that the specificity decreases when the window length is large. On the other hand, the precision also is impacted by the window length, but it seems more stable in the case of KL and Hellinger metrics than in the case of Wasserstein distance.

Let us note that from previously described figures, Hellinger distance seems less competitive than the other two metrics. For illustrative reasons, the $\alpha$ threshold used for creating the mentioned curves was empirically tuned to obtain 20 drifts during the whole stream. Figures 1.5 and 1.6 show results over different threshold values $\alpha$. From Fig. 1.2 and Fig. 1.4 we see a minor difference between KL and Wasserstein dissimilarities. Then, we also present a specific comparison between KL-divergence and Wasserstein distance for different $\alpha$ thresholds using BAC and F1-score values. We fixed the window length to 500 instances (a value that both metrics perform "pretty well" according to Fig. 1.2 and Fig. 1.4). Fig. 1.5 presents two graphics with BAC results, and Fig. 1.6 has two figures with

a comparison between KL and Wasserstein using F1-scores. We appreciate a slight difference between both metrics from the obtained results, indicating that KL-divergence has a better global performance.

It also seems that KL is more robust, i.e., it is less sensitive to the window length and the $\alpha$ value. In addition, KL-divergence is faster than the computation of Wasserstein distance.

***Experimental protocol and implementation.*** We used *python* v3.9, the libraries *numpy* v1.19.5, *stream-learn* v0.8.16 and *scipy.stats* v1.5.4. We used the *scipy.stats.wasserstein_distance* function for computing the Wasserstein distance; KL-divergence and Hellinger distance were implemented by us based on *numpy* functions.

## 1.5 Conclusions and future work

We presented a drift detection method based on the evaluation of changes over the empirical statistical distributions of the data. The method does not require any assumptions about the data. We show its performances using three well-known dissimilarity metrics over binary data with sudden drifts. We compare the behavior of each of the metrics. It is interesting to note that the KL divergence obtains better results globally, and with it, our proposed detector achieves *good* performance. Further work needs to be done exploring real data to analyze statistical differences between the results and other types of concept drifts. Note that the number of bins grows exponentially with the number of dimensions. Then, the binning strategy has well-known limitations in high-dimensional data. For this reason, we also plan to explore other data descriptors.

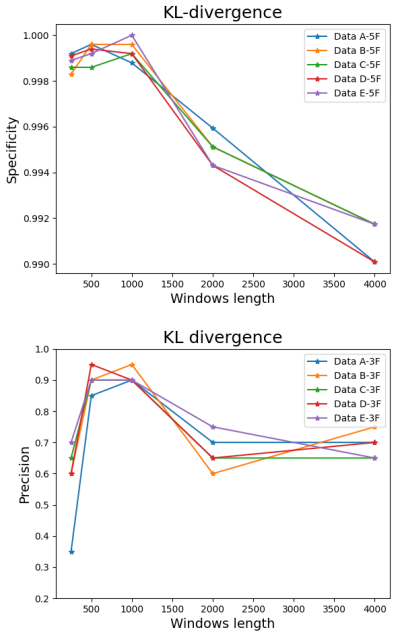


Fig. 1.2: KL-divergence: specificity and precision according to window length.

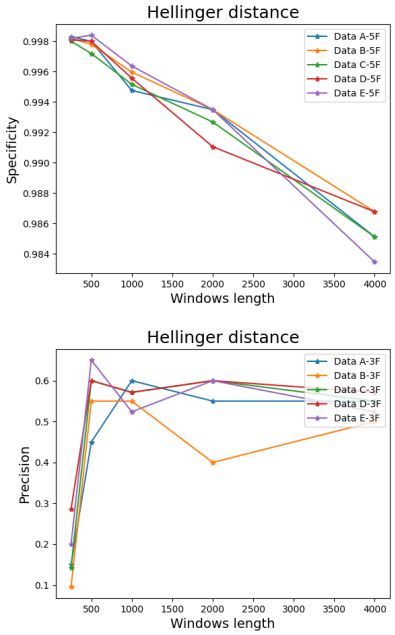


Fig. 1.3: Hellinger distance: specificity and precision according to window length.

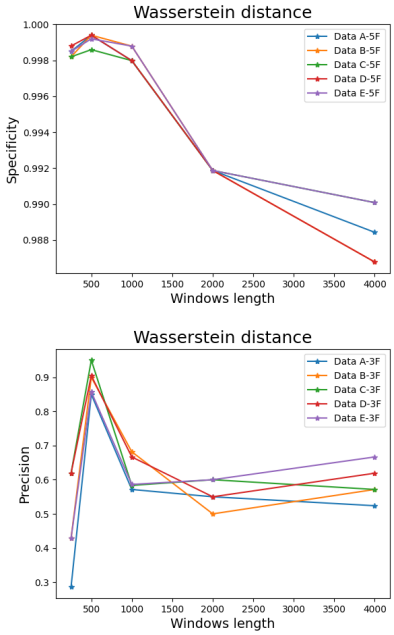


Fig. 1.4: Wasserstein: specificity and precision according to window length.

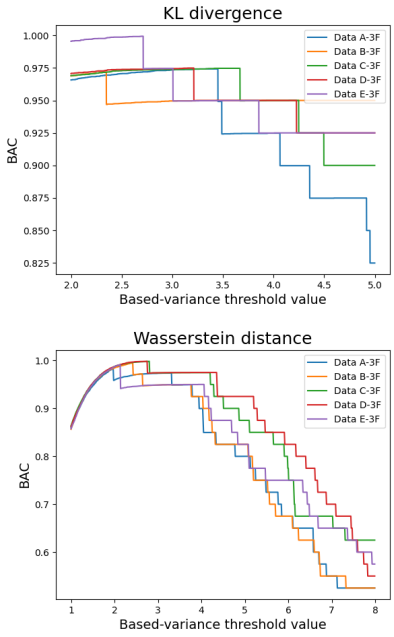


Fig. 1.5: Comparison using BAC between KL-divergence and Wasserstein distance.

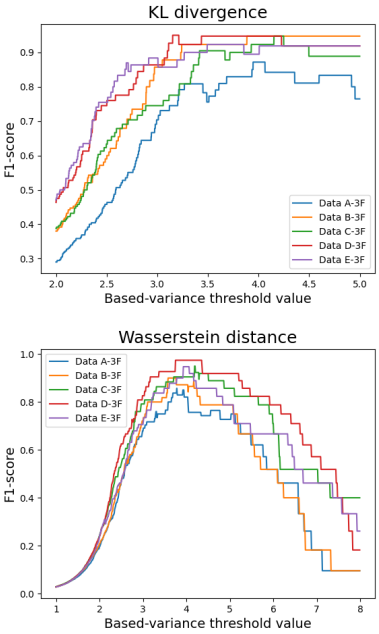


Fig. 1.6: Comparison using f1-score between KL-divergence and Wasserstein distance.

**Acknowledgements** This work was supported by the CEUS-UNISONO programme, which has received funding from the National Science Centre, Poland under grant agreement No. 2020/02/Y/ST6/00037, and the GACR-Czech Science Foundation project No. 21-33574K "Lifelong Machine Learning on Data Streams".

It was also supported by the ClimateDL project (code 22-CLIMAT-02) belonging to the Climate Am-Sud programme, where the central problem is forecasting extreme temperatures in future periods such as in the following summer.

## References

1. Piotr Sobolewski and Michal Woźniak. Concept drift detection and model selection with simulated recurrence and ensembles of statistical detectors. *Journal of Universal Computer Science*, 19(4):462–483, feb 2013.
2. João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, 2014.
3. Fabian Hinder, Valerie Vaquet, and Barbara Hammer. Suitability of Different Metric Choices for Concept Drift Detection, 2022. Available in Arxiv.
4. Sebastián Basterrech and Michal Woźniak. Tracking changes using Kullback-Leibler divergence for the continual learning, 2022. Accepted in IEEE SMC'2022. Available in ArXiv.
5. G. Ditzler and R. Polikar. Hellinger distance based drift detection for nonstationary environments. In *IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*, pages 41–48, 2011.
6. D. Brzezinski and J. Stefanowski. Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1):81–94, 2014.
7. P. M Gonzalvez, Silas de Carvalho Santos, R. Barros, and D. Vieira. A comparative study on concept drift detectors. *Expert Systems with Applications*, 41(18):8144–8156, 2014.
8. Frederik Gustafsson. *Adaptive Filtering and Change Detection*. Wiley, October 2000.
9. Tamraparni Dasu, Shankar Krishnan, and Suresh Venkatasubramanian. An information-theoretic approach to detecting changes in multidimensional data streams. *Interfaces*, pages 1–24, 2006.
10. Kamil Faber, Roberto Corizzo, Bartlomiej Sniezynski, Michael Baron, and Nathalie Japkowicz. WATCH: Wasserstein Change Point Detection for High-Dimensional Time Series Data. In *2021 IEEE Int. Conf. on Big Data (Big Data)*, pages 4450–4459, 2021.
11. Goldenberg Igor and Geoffrey Webb. Survey of distance measures for quantifying concept drift and shift in numeric data. *Knowledge and Information Systems*, pages 591–615, 2019.
12. Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.
13. Alison L. Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *Available in ArXiv*, 2002.
14. Sebastián Basterrech and Pavel Krömer. A nature-inspired biomarker for mental concentration using a single-channel eeg. *Neural Computing and Applications*, 2019.
15. Sebastián Basterrech, Pavel Bobrov, Alexander Frolov, and Dušan Husek. Nature-inspired Algorithms for Selecting EEG Sources for Motor Imagery Based BCI. In Leszek Rutkowski, Marcin Korytkowski, Rafal Scherer, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek M. Zurada, editors, *Artificial Intelligence and Soft Computing*, volume 9120 of *Lecture Notes in Computer Science*, pages 79–90. Springer International Publishing, 2015.
16. P. Ksieniewicz and P. Zyblewski. stream-learn — open-source python library for difficult data stream batch analysis. *Neurocomputing*, January 2022.