A Self-Organizing Clustering System for Unsupervised Distribution Shift Detection

Basterrech, Sebastián; Clemmensen, Line; Rubino, Gerardo

Published in:
Proceedings of the 2024 IEEE International Joint Conference of Neural Networks

Publication date:
2024

Document Version
Peer reviewed version

A SELF-ORGANIZING CLUSTERING SYSTEM FOR UNSUPERVISED DISTRIBUTION SHIFT DETECTION

(RECENTLY ACCEPTED MANUSCRIPT AT IJCNN'2024)

Sebastián Basterrech and Line Clemmensen
Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark
{sebbas,lkhc@dtu.dk}

Gerardo Rubino
INRIA Rennes – Bretagne Atlantique, Rennes, France
Gerardo.Rubino@inria.fr

ABSTRACT

Modeling non-stationary data is a challenging problem in the field of continual learning, and data distribution shifts may result in negative consequences on the performance of a machine learning model. Classic learning tools are often vulnerable to perturbations of the input covariates, and are sensitive to outliers and noise, and some tools are based on rigid algebraic assumptions. Distribution shifts are frequently occurring due to changes in raw materials for production, seasonality, a different user base, or even adversarial attacks. Therefore, there is a need for more effective distribution shift detection techniques.

In this article, we introduce a self-organized continual learning framework designed for monitoring and detecting distribution changes. We explore the problem in a latent space generated by a bio-inspired self-organizing clustering and statistical aspects of the latent space. In particular, we investigate the projections made by two topology-preserving maps: the Self-Organizing Map and the Scale Invariant Map. Our method can be applied in both a supervised and an unsupervised context. We construct the assessment of changes in the data distribution as a comparison of Gaussian signals, making the proposed method fast and robust. We compare it to other unsupervised techniques, specifically Principal Component Analysis (PCA) and Kernel-PCA. Our comparison involves conducting experiments using sequences of images (based on MNIST and injected shifts with adversarial samples), chemical sensor measurements, and the environmental variable related to ozone levels. The empirical study reveals the potential of the proposed approach.

Keywords Distribution shift · Continual Learning · Topology-preserving methods · Self Organizing Maps · Learning Representation · Dimensionality Reduction

1 Introduction

Robustness to data distribution shifts is a crucial design goal when developing foundation models that can be adapted to different machine learning (ML) applications. Classic ML approaches are still vulnerable to perturbations of the input data, and a majority of models have rigid algebraic assumptions and requires i.i.d stationary data samples [1]. As a consequence, distribution shifts in the input datasets can drastically affect the model's performance. The continual learning (CL) paradigm refers to the concept of learning a model sequentially without forgetting previously acquired knowledge, which in the context of analyzing non-stationary data has significant relevance.

In our paper, we focus on a Neural Network (NN) solution, combining the CL context with the use of non-linear dimensionality reductions based on self-organizing methods. We focus on Self-Organizing Maps (SOMs) for this reduction, and compare it to Scale-Invariant Maps (SIMs) (see Section II). To make the proposed framework fast and efficient, we combine the SOM dimensionality reduction with a statistical analysis. This procedure allows us to evaluate

differences between probability distributions in a very efficient way. Moreover, we don't need to assume anything about the distribution of the data. We approximate the distribution of the latent space provided by our self-organizing clustering method through its first moments. The efficiency in detecting changes is based on the fact that our final monitoring signal can be reasonably assumed to be Gaussian, and on the use of the Kullback-Leibler divergence, which is very fast to evaluate in the Gaussian case.

Contributions. We address the distribution shift problem in a context of high dimensional streaming data. The highlights of our contributions are the following: (i) We develop a bio-inspired self-organizing clustering system for assessing distribution changes in data streams. The framework can be applied in unsupervised contexts. (ii) We investigate SOM and SIM projections, which belong to a specific family of non-linear dimensionality reduction techniques. These methods are based on a topology-preserving map, and we explore statistical aspects of the latent space. (iii) By construction, the proposed framework generates a univariate signal that under some restrictions (studied in the paper) can be assumed Gaussian.

We explored the validation of our proposal over three continual learning problems. The experimental results to support our proposal are given in Section IV and discussed in Section V. Section VI presents our conclusions.

2 Related work

2.1 Monitoring and detection of data distribution change

Popular approaches to monitor and assess changes in data distributions focus on the predictive accuracy of the classifier [2]. In this context, the performance of various ensemble classifiers has also been studied in [3–6] The process consists in designing a classifier, and when the accuracy rate significantly decreases, then it is assumed that the data distribution has changed [7, 8]. However, this requires the continuous availability of ground truth labels. Another family of techniques is based on statistical tests on raw data, such as Smirnov-Kolmogorov tests [9] and non-parametric tests [10–12]. Several works present an approach for monitoring aggregation metrics of the raw data, e.g. Cumulative Sum and Exponentially Weighted Moving Average. These techniques compute aggregated statistical metrics of the data and create an additive model of the first moments [13]. For a more complete overview of the state-of-the-art in the use of data descriptors for shift detection, see [1, 11, 14, 15]. Many distribution shift detection methods rely on the computation of the empirical estimated distributions. Even though density function estimation is a fundamental concept in statistics, the estimation of the probability mass function (pmf) is still a complex task, especially when the data belong to a high dimensional space. The approaches are sensitive to outliers and noise, and density estimation techniques may suffer from the curse of dimensionality. The data distribution shift can take different forms, a common taxonomy includes: sudden, gradual, and incremental shifts [16, 17].

2.2 Dimensionality reduction using self-organizing clustering methods

A common approach for analyzing high-dimensional data is to apply dimensionality reduction (DR) using linear or random projections. However, both projections to the marginals and PCA present limitations for capturing the relevant structure of the original data that can cause data shift [1, 18]. On the contrary, some authors recognize the benefits of using PCA as the DR technique in a multivariate unsupervised context [19, 20]. Other DR techniques, such as Kernel-PCA [21], scale with the number of instances. This scalability challenge makes it difficult to apply them effectively for streaming data analysis. An eventual approach for solving this issue is to control the number of instances in each chunk, but it will increase the hyperparameters of the model [22]. **Self-organizing Map (SOM)**. Also referred to as Kohonen Neural Network is a bio-inspired method, which combines concepts from Hebbian learning, vector quantization, and competitive learning [23, 24]. Often, real-world data have important redundancies and intrinsic correlations between the correlated variables. SOMs are useful because they convert complex relationships between high-dimensional data into simple geometric relationships on a regular lattice (most often, a two-dimensional grid) [25]. SOMs methodology was introduced as a particular case of a Neural Network model. The method is a simple non-linear parametric mapping composed of a multi-layered network with Gaussian activation functions. The canonical case can be seen as a two-layered network, with the second layer generally referred to as the *competition layer* or *feature map*. Each neuron in the feature map is characterized by a reference weight vector, that has the same dimension as the input space. In spite of its simplicity, the SOM algorithm is useful for DR and visualization of high-dimensional data [26]. In addition, the method works well for unsupervised problems, and has the property of preserving the most important topological features of the reference data [23, 24, 27]. The network architecture is different from the classic feedforward case. Here, neurons are arranged on a grid. For each neuron i in the feature map, there is a weight vector $\mathbf{w}_i^{(t)} = (w_{i1}^{(t)}, \ldots, w_{id}^{(t)})$ that connects an input pattern $\mathbf{x}^{(t)}$ with neuron i. The weights are adjusted using a physiological interpretation that considers lateral neural inhibition [27–29]. For simplicity, we have here omitted the

reference index for the chunk. The algorithm is iterative, composed of two main phases. The first phase is a competitive learning procedure, that has the goal of finding the neuron that better *represents* the current input pattern. The notion of representation is defined by the concept of closeness among multidimensional points, where all the features of the input data have the same relevance. The representative neuron is the unit in the feature map with associated weight vector closest to the current input sample. At each time t, in the competitive learning step, an input pattern $\mathbf{x}^{(t)}$ is presented to the network and a competition between neurons determines the *representative neuron* $v^{(t)}$, defined by

$$v^{(t)} = \arg\min_i \|\mathbf{x}^{(t)} - \mathbf{w}_i^{(t)}\|. \tag{1}$$

In the canonical SOM, the Euclidean space was selected [29]. The second phase is a regression step, where the update rule for neuron i in the feature map is

$$\Delta\mathbf{w}_i^{(t+1)} = \eta^{(t)} h_v^{(t)}(i)\|\mathbf{x}^{(t)} - \mathbf{w}_i^{(t)}\|. \tag{2}$$

In this rule, $\eta(t) \in [0, 1]$ is the learning rate [30] and the function $h_v^{(t)}(\cdot)$ is a smoother neighborhood kernel of the representative neuron v [23]. Neighborhood functions most often are chosen from the exponential family. We include the notation for time in the neighborhood function, because a common practice is to decrease the diameter of the Gaussian with the number of iterations. Its role is to weight the region of a local neighborhood centered around the representative neuron v, and to control the radius of these balls. A typical choice is the Gaussian radial basis function $h_v(i) = \exp\{-\|r_v - r_i\|/2\sigma^2\}$, where r_v and r_i denote the location vectors of neurons v and i, and σ defines the width of the kernel. For convergence of the regression step, the limit of the kernel $h_v(i)$ is zero with the number of iterations [29]. Another widely selected option for a neighborhood function is the Difference of Gaussians function [31, 32].

Scale-Invariant Map. Some years after SOM, another self-organizing clustering named Scale Invariant Map was introduced [23]. SIM is also a two-layered NN, but with negative feedbacks in the activations [23]. It was shown that this network can use a simple Hebbian learning rule for updating its weights. Early works on SOMs applied Hebbian learning, but the resulting maps were too sensitive to the initial global parameters and showed unstable behavior [29]. More recent SOM specifications mitigated such behavior with the Riccati-type learning equation [29]. SIM seems to be a good alternative to reduce the dimensionality of the data using an essentially Hebbian learning law [23]. The SIM algorithm ignores the magnitude of each correlate input. The process responds only to the relative proportion of the magnitude of the coordinates of the input patterns [23]. Another difference between SIM and the vanilla version of SOM is that before the weights are updated, in SIM the activations pass forward and backward through the neural network. Two possible criteria exist for selecting the representative neuron. One criterion selects the neuron with the greatest activation. Another criterion is given in Expression (1). In this work, we only investigate the latter. The results presented in [23] show how a Hebbian learning rule leads to the following weight update for node i:

$$\Delta\mathbf{w}_i^{(t+1)} = \eta^{(t)} h_v^{(t)}(i)\|\mathbf{x}^{(t)} - \mathbf{w}_v^{(t)}\|, \tag{3}$$

where v is a representative neuron in the feature map, and $h_v^{(t)}(\cdot)$ is the transfer function (often the same Difference of Gaussians function as in SOMs). Note that, according to Expression (3), the representative neuron has a direct effect over the other weights.

This is a difference with SOM, and impacts the tool in how the map creates the clusters. A trained SOM approximates a Voronoi tessellation of the input space, while SIM makes a mapping forming a kind of "pie chart" where each neuron represents a slice of the input data [33].

3 Methodology

3.1 Problem formulation

CL in real-world applications is usually associated with time-dependent problems in a non-stationary environment [34]. Therefore, we incorporate time indexed over a discrete set $\mathcal{T}$ (typically, a segment of integers) into our notation. In this context, we consider a data stream as an ordered sequence of chunks $\{S_1, S_2, \ldots, S_i, \ldots\}$, where chunk S_i consists of a finite data sequence $S_i = \{\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \ldots, \mathbf{x}_i^{(K)}, \ldots\}$ containing realizations of a probability measure, valued in some data space $\mathcal{X}$. We assume that we have a Markov kernel process from $\mathcal{T}$ to $\mathcal{X}$ that associates a distribution p_t of $\mathcal{X}$ with every time point $t \in \mathcal{T}$. A data shift occurs when there exist at least two time stamps t_i and t_j such that p_{t_i} and p_{t_j} are different enough [35]. The goal in a data distribution shift detection problem is to identify all points in time t in $\mathcal{T}$ such that p_t and $p_{t+\Delta}$ differ significantly over a small-time interval $\Delta > 0$. The magnitude of such difference can be estimated using a dissimilarity metric between the underlying distributions [1]. Once the shift is quantified, then a decision rule is applied for deciding if the dissimilitude is either significant or not.

3.2 Unsupervised dimensionality reduction using topographic maps

Comparing probability distributions is a fundamental and difficult problem in statistics, and it is particularly challenging in large dimensions. Therefore, several shift detection techniques in the unsupervised context rely on projecting the data into a latent space. Typical choices are projections over the coordinate axes (marginals), random projections and projections onto the principal components [1]. Linear descriptors of the data present well-known limitations [20]. Other nonlinear projections, such as Kernel-PCA [22], autoencoders, and t-SNE [36], reduce the dimensionality of the data while preserving the reference structure. However, they may present some disadvantages for being used for streaming data analysis. Kernel-PCA scales with the number of samples. Large NNs and autoencoders require additional time windows, and the number of required samples grows with the number of dimensions.

Here, we investigate another approach that consists of using a trained self-organizing clustering method to make the transformation from the input space to the latent space. In the empirical analysis, we study both SOM and SIM techniques. The training phase is made using an initial time-window of the data stream. Then, we continue the learning as usual in a CL scenario. Figure 1 presents a basic approach for tracking changes in the distribution shift. This strategy is based on the assumption that performing a similarity analysis directly in the input space may be too expensive. Consequently, it is more computationally efficient to transform the data in a latent space and, only then, to make the similarity analysis.

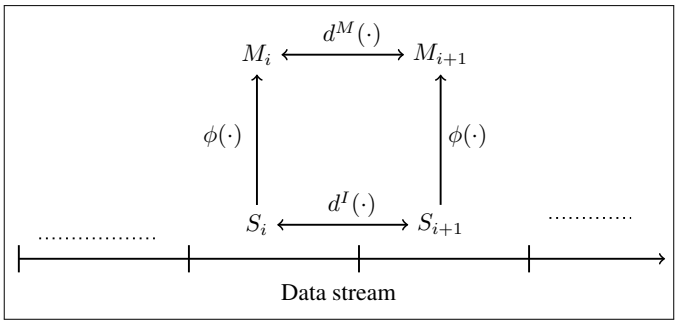

Figure 1: Visualization of the building descriptor of the data. The distance $d^I(\cdot)$ computes a similarity between two distributions directly from the raw data. On the other hand, the distance $d^M(\cdot)$ computes a similarity between two distributions in the latent space. We denote the non-linear projections using a topographic map $\phi(\cdot)$.

3.3 Embedding procedure for the distance matrix

Both SOMs and SIMs have frequently been applied as an unsupervised tool for clustering problems [29]. Also, they showed to be effective for DR. Both methods project the input into the feature map, which it is most often a 2-dimensional space, composed of the coordinates of the neurons. Therefore, it is a powerful reduction that can be useful in many applications, but for detecting distribution changes in the input space such a dramatic reduction of the original information may have negative consequences. In addition, projection over coordinates has other drawbacks. Coordinates are arbitrarily selected, and more often they don't consider any property of the data itself. There are even problems in cases where the coordinates are not natural in any sense [37].

To overcome these difficulties, we decided to use a less rigid projection that is coordinate-free and also that contains more information from the original space. Instead of projecting over two dimensions, we propose a reduction to a latent space in p^2 dimensions, where p is the number of neurons in the feature map. Hence, our focus is on the geometric properties of the latent space. We denote by $\phi : \mathcal{X} \to \mathcal{H}$ the projection function made by the self-organizing clustering method, where $\mathcal{H}$ denotes the latent space with p^2 dimensions. Once the model is trained, we construct a matrix D, where the element in position (i, j) is the distance between the input data and the neuron located at (i, j) position on the lattice grid. In other words, $d_{ij} = \|\phi(\mathbf{x}) - \mathbf{w}_k\|_2$, where $\mathbf{w}_k$ is the weight vector associated with the (i, j)-neuron. For computing distances among low-dimensional points in the latent space, we use the Euclidean metric. In spite of the previous reduction in dimension to value p^2, this is in general still large. Note that, standard empirical experiments use a squared grid of few hundred of neurons [24].

With the nonlinear projection of the topographic map, the dimension is reduced from d to p^2, which may still be too large for estimating densities. Then, we apply an additional dimensionality reduction step by moving to the first moments of the distribution represented in D. The first moments of a random sequence capture different aspects of the probability distribution that are useful to approximate it [38]. To evaluate the relevance of the moments in our

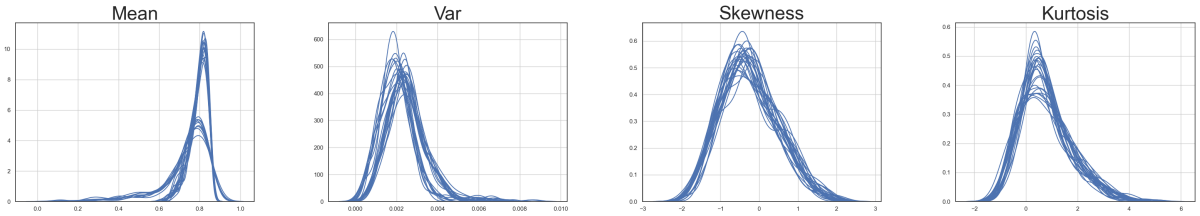

Figure 2: Visual comparison of the significance of each of the moments for representing the information in the matrix D. The data corresponds to the problem of MNIST with adversarial samples in a CL setting.

problem, we empirically analyzed the impact of the first four moments. Let us denote $\mathbf{m} = (m_1, m_2, m_3, m_4) \in \mathbb{R}^4$ these moments, where m_1 is the mean, m_2 the variance, m_3 the skewness, and m_4 the kurtosis. Figure 2 presents an example of the information provided by each of the first four moments. The analysis was made offline, comparing the samples from the MNIST benchmark data versus the adversarial MNIST samples (for a description of the benchmark data see Section 4.1). Figure 2 has four plots, each having information about the mean, variance, skewness and kurtosis. Each curve represents the estimated pmf of the sequence $\mathbf{m}_i^{(1)}, \ldots, \mathbf{m}_i^{(K)}$, for all chunks S_i. According to this chunk-based analysis, it is visible that the left plot in Figure 2 distinctly reveals two families of densities. One group of curves represents the pmf calculated using the original MNIST samples, while the other group corresponds to the pmf computed with the adversarial MNIST samples. This shows that the mean captures *sufficient* information from matrix D to be able to distinguish groups of data distributions. Therefore, the proposed framework includes a final dimensionality reduction step, which involves transforming the entire matrix D into a univariate sequence. Then, we represent D using the first moment (mean function) of its p^2 elements (seen as p^2 independent samples of an auxiliary random variable).

3.4 Workflow of the proposed framework

Figure 3 illustrates the main three-stages of the proposed framework for monitoring distribution variation in the data stream:

(i) **Dimensionality reduction**. A non-linear projection of the high-dimensional input pattern is applied. The projection is made using a topology-preserving mapping.

(ii) **Clustering analysis**. A matrix is created that displays the geometric relations between the projected points and their distances to the representative neurons of each cluster. Note that, this study is made in the latent space that has much lower dimensionality than the original space.

(iii) **Statistical summary**. The distance matrix is embedded using a statistical summary. In this work, we present experiments with the matrix D summarized with the mean function. Then, the framework provides a univariate signal for monitoring the changes in the distribution.

(iv) **CL step - Model update**. Each representative neuron is characterized by its weight vector, that is trained using an initial window of samples. In case of detecting a significant change in the distribution, the weights are updated with the data presented in the last seen chunk.

3.5 Quantification of the distribution shift

Observe first that the distribution of the mean of a sequence, as the number of terms increases, tends toward a Gaussian under conditions often satisfied in practice (Central Limit Theorem, CLT). We need enough terms (a few dozens is in general enough) and independence between them, or at least a weak correlation. Assuming this holds will simplify the computations that we describe in the following lines. The number of terms for satisfying the CLT is given by the number of neurons in the SOM/SIM lattice. We compare the data distributions in two consecutive chunks S_i and S_{i+1}, using the Kullback-Leibler (KL) divergence. The KL-divergence method has been employed to monitor alterations in the data distribution [39]. As illustrated in Figure 1, the KL divergence score is calculated in the projected space, rather than making the assessment directly on the original data. Given two pmfs p and q, defined in a common data space χ, the KL-divergence from p to q is [40]

$$\mathrm{KL}(p \parallel q) = \sum_{s \in \mathbb{S}} p(s) \log \frac{p(s)}{q(s)}. \tag{4}$$

This is not strictly a distance. It does not satisfy the triangular inequality and is not symmetric in inputs. The $\mathrm{KL}(\mathrm{p} \parallel \mathrm{q})$ quantifies the information lost if we use q as an approximation of p. Even though the KL-divergence is not strictly a distance, it has several useful properties and advantages over mathematical distances (for more details, see [40–42]). One of the benefits of KL is that there exists a relationship with the expected value of the likelihood ratio. Moreover, Expression (5) takes a specific form for specific distributions. This is the case when two Gaussian distributions are compared. Let μ_p and σ_p^2 (resp. μ_q and σ_q^2) be the mean and variance of the pmf p (resp. q). In this case, we have

$$\mathrm{KL}(p \parallel q) = \log \frac{\sigma_q}{\sigma_p} + \frac{\sigma_p^2 + (\mu_p - \mu_q)^2}{2\sigma_q^2} - \frac{1}{2}. \tag{5}$$

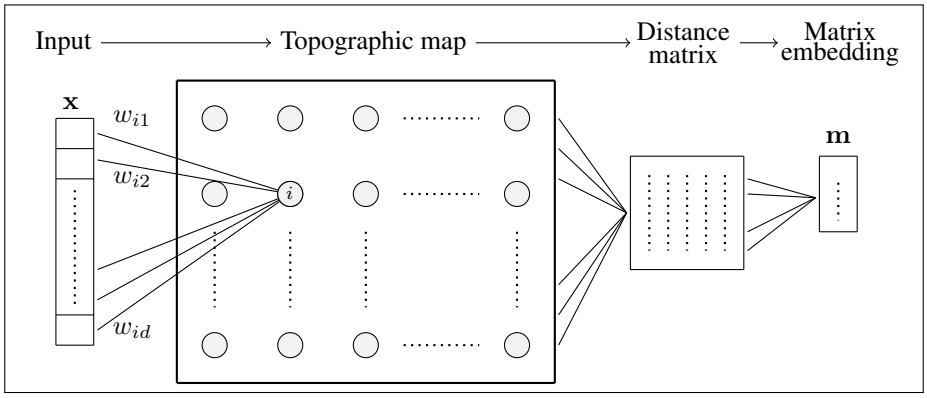

Figure 3: Visualization of the proposed approach.

3.6 Decision rule module

The framework has an additional independent block with the decision rule. Here, we work with a simple and fast procedure taken from the area of time series analysis for detecting outliers. Let $\{l_1, l_2, \ldots\}$ be the signal with the KL-divergence scores. This signal is computed as described in Section 3.5. We apply a rule that is often used in EEG analysis to detect artifacts [43]. We define a *critical* point location (i.e., timestamps when the distribution change is relevant) when $l_i \notin [\bar{l} - \alpha\sigma, \bar{l} + \alpha\sigma]$, where $\bar{l}$ denotes the mean of the sequence, σ is the standard deviation, and α is a real-value control parameter.

4 Results

4.1 Datasets

There is a lack in the community of a large and diverse collection of real data streams in a high-dimensional space [17], especially in unsupervised streaming data analysis. As a consequence, we generated a synthetic data stream with injected drifts. The created stream has samples from MNIST [44] and adversarial samples of MNIST data generated in [15]. Furthermore, we carried out experiments over other two real-world datasets.

MNIST with adversarial samples [15]. We created a data stream, emulating recurrent data shifts. It contains original MNIST images and adversarial images created via FGSM. For more details about the adversarial MNIST data see [15]. Figure 4 depicts how the streaming data was generated. The construction of the data stream simulates changes in distribution when the datasets are exchanged from original samples to fake samples. A similar method of data stream generation was used in the context of fake information classification [45]. It has inter-exchanged samples from both datasets. Every 2000 images, we exchange the datasets. We assume that an efficient data shift monitoring will detect the time-stamps where there are changes between regular MNIST samples and adversarial MNIST samples.

Gas sensor data stream [46]. This dataset was initially proposed as a classification benchmark problem with 16 sensors monitoring metal-oxide gas over three years, and where sensor drifts occur and deteriorates classifications. During the last years, the dataset has become popular in the domain of concept drift detection. Here, we analyze an updated version with 128 features and six classes representing gaseous substances [17, 46]. The number of changes is 38.3 shifts according to [47].

Ozone data stream [17]. This dataset contains air measurements collected from 1998 to 2004. It has 72 features and a binary output variable (ozone day and normal day). The problem has only 2534 samples, for more details see [17, 48].

It is considered a challenging problem because the data is imbalanced. In addition, it *seems* that they have a high frequency of distribution changes [17].

4.2 Experimental setup

The three studied benchmark problems can be analyzed in a supervised context. However, here we make the monitoring of the data distributions only by analyzing the covariate variables. For each of the problems, we used the first 30% of samples for training the parameters of the self-organizing clustering methods (SOM and SIM). This training was made offline, as a pre-phase of the continual learning process. Then, we simulated a streaming environment using the `scikit-multiflow` package [49]. Both SOM and SIM have a grid with 10×10 neurons. The two parameters of the decision rule, α and window size, were analyzed on a grid, α in $[1, 29]$ and window size in $[1, 25]$, only considered the even values.

We generated 30 data streams using the MNIST and adv. MNIST datasets. The chunk size parameter was studied in the set $\{50, 100, 200, 500\}$. For the MNIST problem, in case of a chunk size equal to 200, the stream has simulated sudden drifts. The other chunk values create chunks with original MNIST data and fake samples (both in the same chunk).

Metric choices and baseline. As a way to assess the quality of detecting distribution shifts, we compute the Kappa score for the generated MNIST data stream. The Kappa score is commonly used to compare binary sequences. The data stream was synthetically created, so we have the precise time-stamps wherein the shifts occur. We also use the same MNIST data to evaluate PCA and Kernel-PCA as drift detector tools. We use those two classic clustering techniques as a baseline. We assess the capacity of PCA and Kernel-PCA as follows. We utilize an initial time window comprising 30% of the data stream for training both clustering methods. Subsequently, we apply the trained methods to project the data points into a latent space, following a CL setting. Once the data in the current chunk are projected, a pmf is estimated using the projected points. Subsequently, comparisons between pmfs were conducted in the latent space using two popular metrics [1]: the cumulative histogram and the Kolmogorov-Smirnov test. The monitoring of the magnitude of the shifts was done with KL-divergence.

4.3 Empirical evaluation

Figure 5 presents the monitoring of the MNST data stream. There are 30 curves per graph, with each curve representing the results from different data streams. The left curve displays the results when the number of samples in the chunk was 100 (with injected shifts occurring at time steps 20, 40, 60, 80, 100, and 120). Conversely, the right curve illustrates the results when the number of samples was 200, with injected shifts in time-stamps $\{10, 20, 30, 40, 50, 60\}$. Both figures show how the proposed framework effectively monitors changes in the distribution. Contrarily, Figure 6 illustrates the similarity between two consecutive chunk distributions when the baseline methods (PCA and Kernel-PCA) were applied (with a chunk size equal to 200). The left figure presents the results of assessing the difference between the cumulative histograms in the latent space (with four principal components). The right figure presents the results of applying the Kolmogorov-Smirnov test to measure the discrepancy between the cumulative histograms. Both graphics show the complexity of the problem, and how PCA and Kernel-PCA have limitations for detecting fine types of drifts in the data.

In addition, both baseline techniques have the additional computational cost of needing to compute the pmf estimation. This computation is avoided in our framework because of the framework's construction; the monitoring signal follows a Gaussian distribution. Figure 7 visualizes the impact of the two parameters with respect of Kappa score in the decision rule. The left graphic depicts results obtained by SOM. The right graphic has the results obtained by SIM. The Kappa score is indeed a metric often used in the context of binary classification, it ranges between -1 and 1. We can see that SOM remains stable even when parameters change, consistently achieving higher Kappa values compared to those obtained by SIM. Figure 8 presents an additional comparison between SOM and SIM. In this scenario, the results were obtained from a data stream containing chunks with samples from both the original MNIST dataset and the adversarial MNIST dataset. Consequently, this problem is more complex than monitoring streams, where the shift is precisely presented at the moment of connecting consecutive chunks. The figure shows the different impacts of three values of α (specifically, 3, 10, 15) with a fixed window size of 8. The framework utilizing SOM achieves high accuracy. We visualize the results of the Gas sensor data problem in Figures 9 and 10. Since this is real-world data, the shifts are not explicitly labeled. However, according to the literature, there are approximately 38 shifts [17].

The graphics also aim to demonstrate the framework's sensitivity to the parameters of the α-decision rule. Figures 11 visualize the monitoring signal when the Ozone level dataset is analyzed. This real-world problem is particularly challenging because the data is imbalanced, there are few samples, and there are many features relative to the number of samples. According to [17], this data stream exhibits numerous shifts (over 90 shifts). We can see that both methods

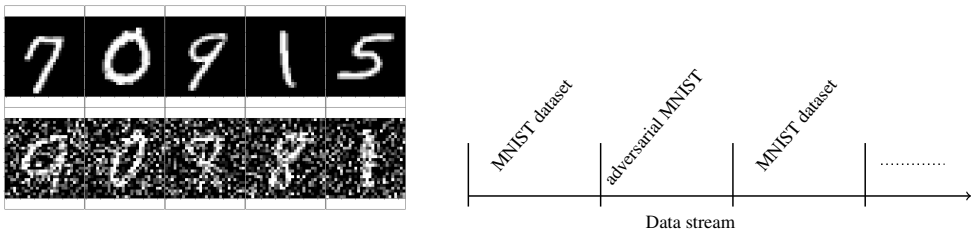

Figure 4: MNIST problem with adversarial samples: This example illustrates the transition between the sequence of images before and after the injected drift. The second row of images contains the sequence of adversarial samples.

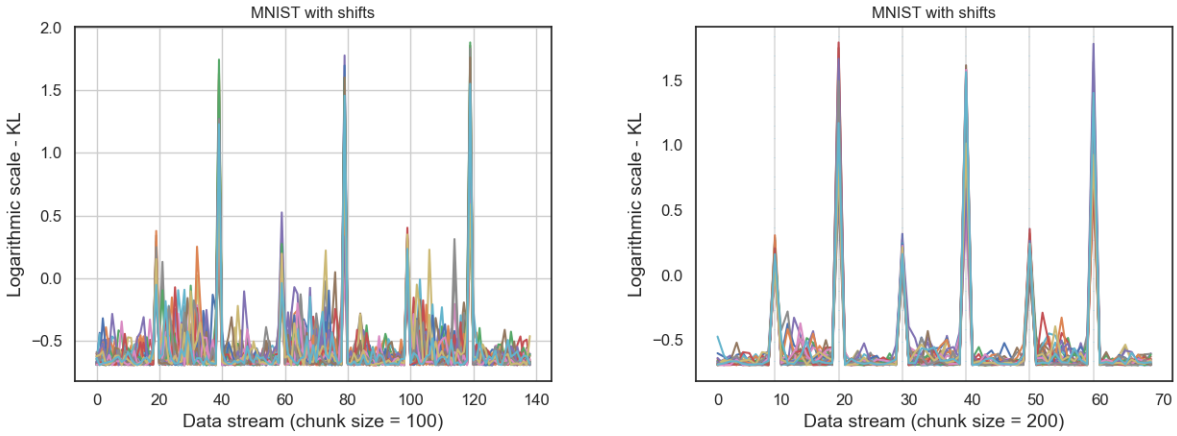

Figure 5: Analysis of distribution shifts with fixed reference time windows: Each curve represents the shift monitoring for a specific generated data stream. We generated 30 data streams using the MNIST dataset and injected shifts using adversarial samples. The left figure was generated with data streams using chunk size 100 samples. The right figure was generated with data streams using chunks with 200 samples.

are able to assess clearly discrepancies between distributions. However, it appears that the parameters chosen in the $\alpha-$decision rule strongly impact the decisions, whether they indicate a shift or not.

5 Discussion

Gaussian assumption. One critical aspect of our proposal is the robustness of the Gaussian assumption, needed to support the simplified computation of the KL-divergence in that case (see Subsection 3.5). This relies on the Central Limit Theorem. In our case, the number of terms is given by the size of the grid of neurons, which is large enough regarding standards. The weak correlation between terms is more tricky, but the assumption often holds. This is an issue to be explored further in future studies.

Selected decision rule for the detection. Note that, the parameters of the decision rule impact in the monitoring signal too. Because when the algorithm considers that a shift occurred, we update the weights of the topographic map. This has consequences in the monitoring signal. However, according to the experiments, the monitoring signal is not too much sensitive to the decision rule parameters, as it is the shift detector (given by the decision rule module). Other decision rule techniques should be explored. Threshold over peak is the easiest rule that can be integrated. The monitoring signal is univariate, therefore another possibility is to apply online outlier detection techniques.

Computational costs Note that the proposed framework is composed of a non-linear projection (two layered NNs), a computation of a squared distance matrix (10×10 in our experiments), these distances are computed in the latent space (that it is relatively low), and the parametric estimation of KL-divergence (it requires the mean and standard deviation of a sequence with the number of samples equal to the length of the chunk). As a consequence, the method doesn't have the cost of computing histograms in large spaces that often appear in other techniques (e.g. moment tree, marginal bins [11]). The method doesn't require either the computation of an inverse matrix as in the Kernel embedding

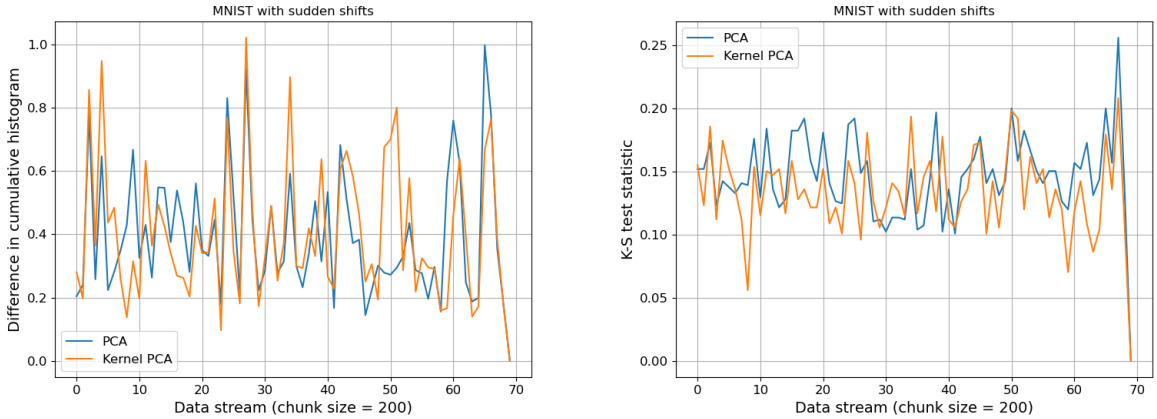

Figure 6: Example of the difficulty of the problem. The projection was done with PCA and Kernel-PCA. The left figure shows the difference between two cumulative histograms computed over two consecutive chunks.The right figure shows the statistic of Kolmogorov-Smirnov test over two consecutive chunks.

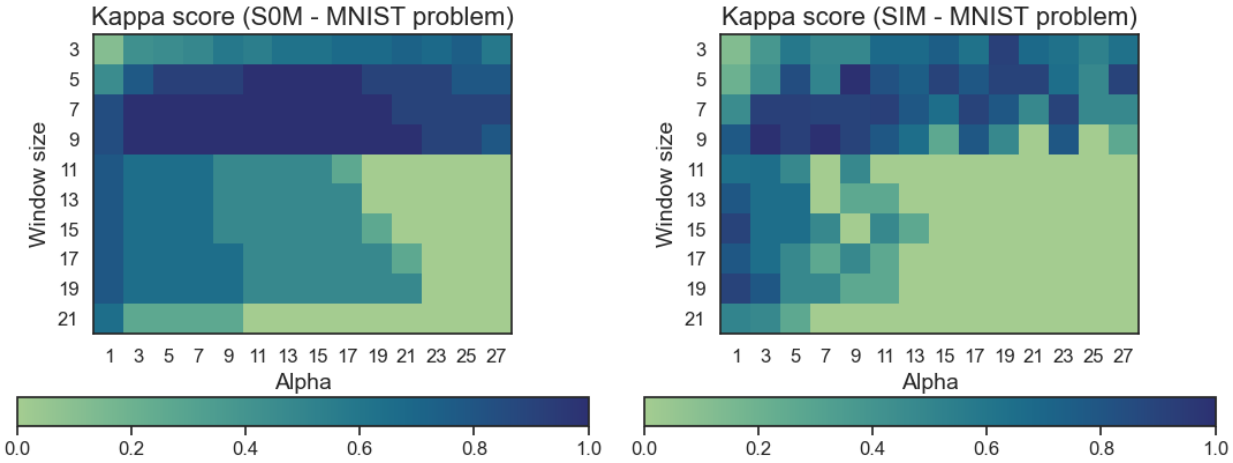

Figure 7: Analysis of sensitivity regarding the two parameters used in the decision rule (the α rate and the window size) was conducted. The accuracy of shift prediction was assessed using the Kappa score.

method [1]. Furthermore, the distance matrix has fixed dimensions and doesn't scale with the number of samples in the chunk.

6 Conclusions and future work

We proposed a general system for assessing distribution shifts in high-dimensional streaming data. The approach is based on a bioinspired nonlinear projection with the property of preserving topological characteristics of the reference data, and a statistical representation of the latent space. The proposed method does not have any assumption about the distribution of the reference data stream. In addition, it can be applied to both supervised and unsupervised problems. We empirically study the performance of our method on three problems, and we see how the proposed system with a low budget can easily monitor the distribution changes of the input data. Furthermore, we contrast the results of our framework with PCA and Kernel-PCA on an annotated dataset. The results are promising and show how our proposal produce a much clearer monitoring signal than these classic techniques. We believe that the presented work offers a significant step toward the application of topology-preserving maps in the domain of distribution shifts.

In the near future, we plan to include other topology-preserving mappings (e.g. generative topographic mapping), and to explore the incorporation of other techniques for analyzing the monitoring signal.

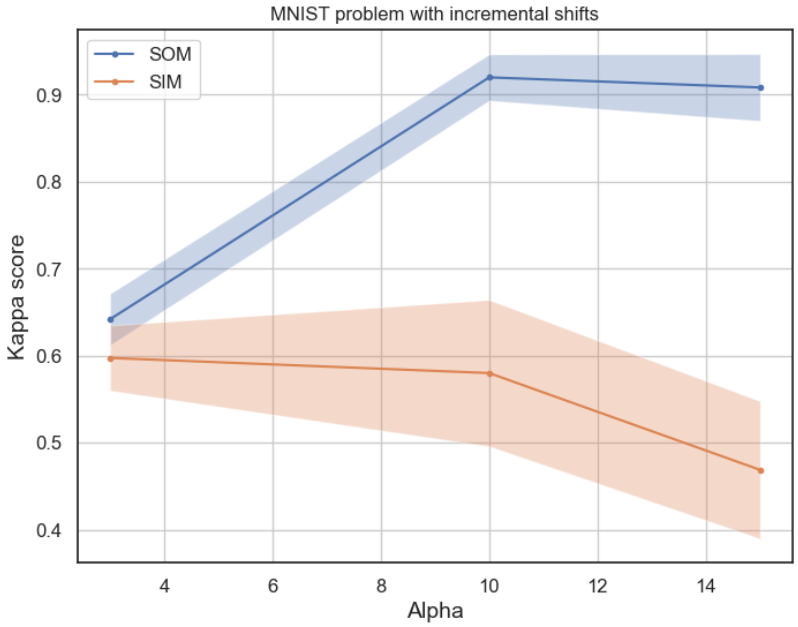

Figure 8: Incremental shift. Analysis of sensitivity of the two parameters used in the decision rule. The data stream has the MNIST with adversarial samples, and the shift is injected incrementally. The CI was computed with the results of 30 different generated data streams.

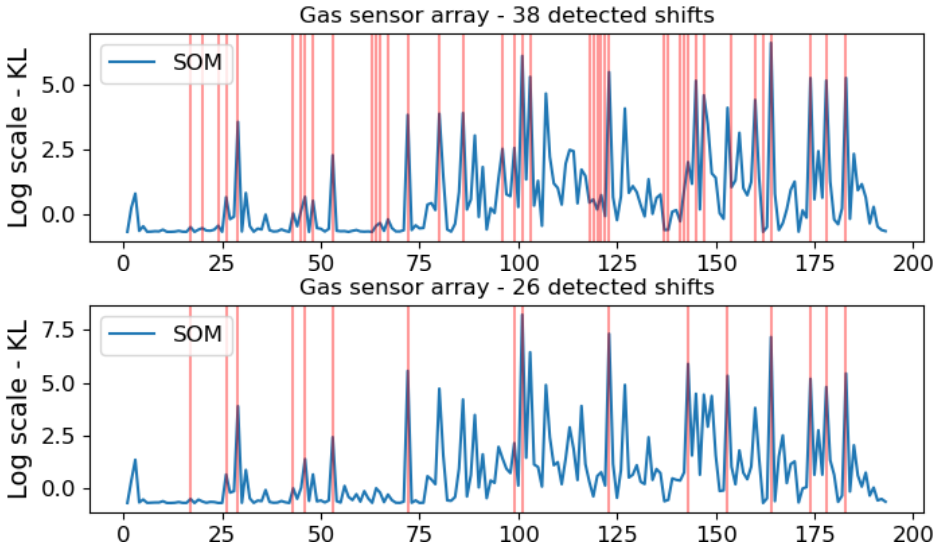

Figure 9: Gas sensor array problem. Monitoring with SOM. The figure at the top has parameters of control $\alpha = 8$ and windows size equal to 10. There are 38 detected drifts, what it is the same as the solution described in [17]. The figure in the bottom has parameters of control $\alpha = 15$ and windows size equal to 10.

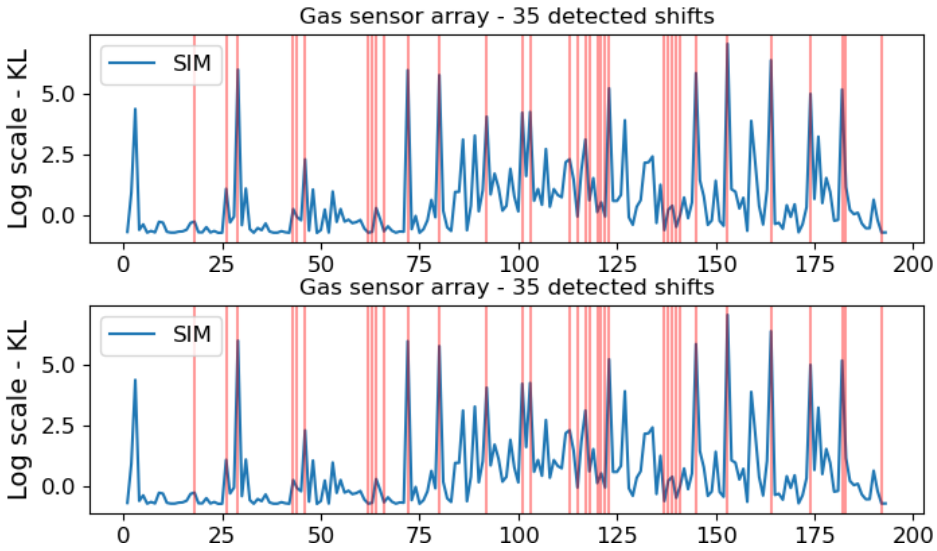

Figure 10: Gas sensor array problem. Shift monitoring with SIM. The figure in the top has control parameters $\alpha = 7$ and windows size equal to 10. There are 35 detected drifts. The figure in the bottom has parameters of control $\alpha = 15$ and windows size equal to 10. Red vertical lines show the time-stamps where our algorithm considered a significant distribution change.

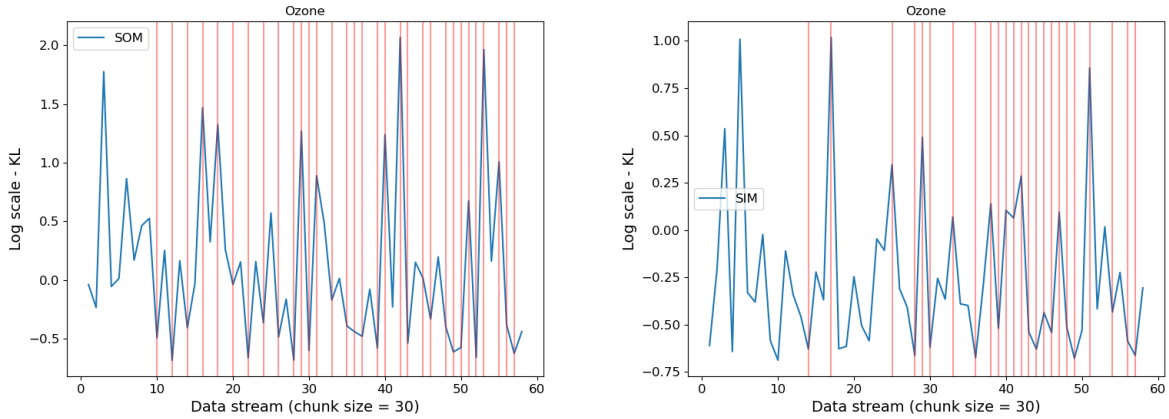

Figure 11: Ozone data set. Shift monitoring with SOM and SMM. The left figure has parameters of control $\alpha = 3$ and windows size equal to 8. The right figure was made with SIM, and it has parameters of control $\alpha = 4$ and windows size equal to 8. Red vertical lines show the time-stamps where our algorithm considered a significant distribution change.

Acknowledgements

This work was supported by the LF Experiment grant number R400-2022-1201, and it was also supported by the 22-CLIMAT-02 project entitled "Using deep learning spatial-temporal graph models for seasonal forecasting of extreme temperature events" belonging to the Climate AmSud programme.

References

[1] Fabian Hinder, Valerie Vaquet, and Barbara Hammer. Suitability of Different Metric Choices for Concept Drift Detection. In Tassadit Bouadi, Elisa Fromont, and Eyke Hüllermeier, editors, *Advances in Intelligent Data Analysis XX*, pages 157–170, Cham, 2022. Springer International Publishing.

[2] Igor Goldenberg and Geoffrey I Webb. Survey of distance measures for quantifying concept drift and shift in numeric data. *Knowledge and Information Systems*, 60:591–615, 2019.

[3] B. I. F. Maciel, S. G. T. C. Santos, and R. S. M. Barros. A lightweight concept drift detection ensemble. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1061–1068, Nov 2015.

[4] J.Z. Kolter and M.A. Maloof. Dynamic weighted majority: a new ensemble method for tracking concept drift. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 123 – 130, nov. 2003.

[5] Lei Du, Qinbao Song, Lei Zhu, and Xiaoyan Zhu. A selective detector ensemble for concept drift detection. *The Computer Journal*, page bxu050, 2014.

[6] Andrzej Lapinski, Bartosz Krawczyk, Pawel Ksieniewicz, and Michal Wozniak. An Empirical Insight Into Concept Drift Detectors Ensemble Strategies. pages 1–8, 07 2018.

[7] Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In *In SBIA Brazilian Symposium on Artificial Intelligence*, pages 286–295. Springer Verlag, 2004.

[8] 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.

[9] Piotr Sobolewski and Michal Wozniak. 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.

[10] Isvani Inocencio Frias Blanco, Jose del Campo-Avila, Gonzalo Ramos-Jimenez, Rafael Morales Bueno, Agustin Alejandro Ortiz Diaz, and Yaile Caballero Mota. Online and non-parametric drift detection methods based on hoeffding's bounds. *IEEE Trans. Knowl. Data Eng.*, 27(3):810–823, 2015.

[11] Fabian Hinder, André Artelt, and Barbara Hammer. Towards Non-Parametric Drift Detection via Dynamic Adapting Window Independence Drift Detection (DAWIDD). In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[12] Line H. Clemmensen and Rune D. Kjærsgaard. Data Representativity for Machine Learning and AI Systems, 2023.

[13] Gordon J. Ross, Niall M. Adams, Dimitris K. Tasoulis, and David J. Hand. Exponentially weighted moving average charts for detecting concept drift. *Pattern Recognition Letters*, 33(2):191 – 198, 2012.

[14] 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.

[15] Stephan Rabanser, Stephan Gunnemann, and Zachary C. Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada*, 2019.

[16] M. Wozniak, Pawel Zyblewski, and Pawel Ksieniewicz. Active weighted aging ensemble for drifted data stream classification. *Information Sciences*, 630:286–304, 2023.

[17] V. M. A. Souza, D. M. Reis, A. G. Maletzke, and G. E. A. P. A. Batista. Challenges in Benchmarking Stream Learning Algorithms with Real-world Data. *Data Mining and Knowledge Discovery*, 34:1805–1858, 2020.

[18] 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.

[19] 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.

[20] Abdulhakim Ali Qahtan, Basma Alharbi, Suojin Wang, and Xiangliang Zhang. PA PCA-Based Change Detection Framework for Multidimensional Data Streams: Change Detection in Multidimensional Data Streams. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[21] Sebastian Mika, Bernhard Scholkopf, Alex Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel PCA and De-Noising in Feature Spaces. In *Neural Information Processing Systems*, 1998.

[22] B. Schölkopf and A.J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning. MIT Press, 2002.

[23] Colin Fyfe. *Hebbian Learning and Negative Feedback Networks*, volume XVIII of *Advanced Information and Knowledge Processing*. Springer-Verlag London, first edition, 2005.

[24] Teuvo Kohonen. Essentials of the self-organizing map. *Neural Networks*, 37:52–65, 2013.

[25] Christos Ferles, Yannis Papanikolaou, and Kevin J. Naidoo. Denoising Autoencoder Self-Organizing Map (DASOM). *Neural Networks*, 105:112–131, 2018.

[26] Ayu Saraswati, Van Tuc Nguyen, Markus Hagenbuchner, and Ah Chung Tsoi. High-resolution Self-Organizing Maps for advanced visualization and dimension reduction. *Neural Networks*, 105:166–184, 2018.

[27] Hujun Yin. *Learning Nonlinear Principal Manifolds by Self-Organising Maps*, volume 60, pages 68–95. 09 2007.

[28] C. Fyfe, M. Peña, and W. Barbakh. Topology preserving mappings for data visualization. In *Principal Manifolds for Data Visualization and Dimension Reduction*, pages 132–152. Springer Heidelberg, 2007.

[29] Teuvo Kohonen. *Self-Organizing Maps*, volume 30. Springer Series in Information Sciences, third edition, 2001.

[30] Leslie N. Smith and Nicholay Topin. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates, 2018.

[31] Thomas Voegtlin. Recursive self-organizing maps. *Neural Networks*, 15:979–991, October 2002.

[32] D. MacDonald and C. Fyfe. The Kernel Self Organising Map. In *KES'2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings*, volume 1, pages 317–320 vol.1, 2000.

[33] D. MacDonald, J. McGlinchey, and C. Fyfe. Comparison of Kohonen, Scale-Invariant and GTM Self-Organizing Maps for interpretation of Spectral Data. In *European Symposium on Artificial Neural Networks*, pages 117–122, 1999.

[34] 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.

[35] Geoffrey Webb, Roy Hyde, Hong Cao, Hai-Long Nguyen, and François Petitjean. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30, 07 2016.

[36] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[37] Erik Carlsson, Gunnar Carlsson, and Vin Silva. An algebraic topological method for feature identification. *Int. Journal of Computational Geometry and Applications*, 16:291–314, 08 2006.

[38] Ismail Chamseddine. *Construction of Random signals from their Higher Order Moments*. Createspace Independent Publishing Platform, Scotts Valley (Californie), 2012.

[39] Sebastián Basterrech and Michal Woźniak. Tracking changes using Kullback-Leibler divergence for the continual learning. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3279–3285, 2022.

[40] Tamraparni Dasu, Shankar Krishnan, and Suresh Venkatasubramanian. An information-theoretic approach to detecting changes in multidimensional data streams. *Interfaces*, pages 1–24, 2006.

[41] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[42] Sebastián Basterrech, Jan Platoš, Gerardo Rubino, and Michał Woźniak. Experimental Analysis on Dissimilarity Metrics and Sudden Concept Drift Detection. In Ajith Abraham, Sabri Pllana, Gabriella Casalino, Kun Ma, and Anu Bajaj, editors, *Intelligent Systems Design and Applications*, pages 190–199, Cham, 2023. Springer Nature Switzerland.

[43] Sebastián Basterrech and Pavel Krömer. A nature-inspired biomarker for mental concentration using a single-channel eeg. *Neural Computing and Applications*, 2019.

[44] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[45] Sebastián Basterrech, Andrzej Kasprzak, Jan Platos, and Michal Wozniak 0001. A Continual Learning System with Self Domain Shift Adaptation for Fake News Detection. In *10th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2023, Thessaloniki, Greece, October 9-13, 2023*, pages 1–10. IEEE, 2023.

[46] A Vergara, S Vembu, T Ayhan, MA Ryan, ML Homer, and R Huerta. Chemical gas sensor drift compensation using classifer ensembles. *Sensors and Actuators: Chemical*, B(166):320–329, 2012.

[47] M. de Souto, R. Soares, A. Santana, and A. Canuto. Empirical comparison of dynamic classifier selection methods based on diversity and accuracy for building ensembles. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1480 –1487, june 2008.

[48] Dimitrios Effrosynidis and Avi Arampatzis. An evaluation of feature selection methods for environmental data. *Ecological Informatics*, 61:101224, 2021.

[49] Jacob Montiel, Jesse Read, Albert Bifet, and Talel Abdessalem. Scikit-Multiflow: A Multi-output Streaming Framework. *Journal of Machine Learning Research*, 19(72):1–5, 2018.