An Adaptative Active Approach with Monitoring Quality of Stream Modeling in the Presence of Concept Drift

Fatemeh Haadji Ahmadi[1], Seyed Taghi Akhavan Niaki[2], Rassoul Noorossana[3], Hamid R. Rabiei [4]

[1] Department of Industrial Engineering, Islamic Azad University, Qazvin, Iran, Email: hajiahmadi@qiau.ac.ir

[2]Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran, Email: Niaki@Sharif.edu

[3] Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran, Email: rassoul@iust.ac.ir

[4]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, Email: rabiee@sharif.edu

An Adaptative Active Approach with Monitoring Quality of Stream Modeling in the Presence of Concept Drift

Fatemeh Haadji Ahmadi[1], Seyed Taghi Akhavan Niaki[2], Hamid R. Rabiei [3], Rassoul Noorossana [4]

[1] Department of Industrial Engineering, Islamic Azad University, Qazvin, Iran, Email: hajiahmadi.f@gmail.com

[2]Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran, Email: Niaki@Sharif.edu

[3] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, Email: rabiee@sharif.edu

[4]Department of Industrial Engineering, Iran University of Science and Technology,Tehran, Iran, rassoul@iust.ac.ir

Abstract

Monitoring the quality of statistical models for stream big data processing in nonstationary environments is a practical approach to provide the adaptability of the modeling process in the face of change and drift. It is cost-effective to offer a well-functional calibrated processing system to maintain high-level accuracy with minimum variance in adaptive modeling. The study of drifting concepts and continuous learning has helped predict the behavior of various real-time big data stream adaptations. A concept drift detection method provides an adaptive mechanism in nonstationary characteristics of providing triggers to adaptation design when the drift is detected. Stream processing is considered a production system with prediction and adaptation output from a quality control perspective. This research provides a concept drift detection method that monitors non-defective production proportion as a critical quality performance indicator. An active adaptive framework with explicit concept drift detection to govern the incremental learning process utilizing triggering mechanisms is an issue that has been focused on in this study. The estimation of the quality indicator originates from the concept of Bayesian analysis in predicting changes in the model parameters. The estimate is updated sequentially then a detection control design is used to trigger the possible evolution of distribution parameters to the adaptation module, providing minor adjustments to passive adaptation. Among the key contributions of this research, the algorithmic method described herein provides superior adaptation among common triggering methods in active detection alongside partial fit in Gaussian Naive Bayes as incremental learning when tested on electricity datasets, the most commonly used in concept drift detector studies. The general framework of this research is a variety of data stream applications in the data stream that have concept drift.

Keywords: Stream data; nonstationary; online learning; adaptive active learning; Concept drift; Conjugate analysis

1. **Introduction**

It is not difficult to name processes and systems that generate a high volume of data in diverse varieties and at high speed these days. The dynamic and unstable nature of real-world applications influences learning techniques through changes in data produced sequentially at high speed. Big data application includes mobile calls, emails, spam prediction sensor networks, financial, climate data and customer clickstreams, Global Positioning System (GPS) data, transmission control protocol (TCP/IP) traffic, and large-scale log data. These examples are likely to continuously produce a vast volume of data from nonstationary distributions.

In a real-time scenario, a data stream includes concept drifts and has a nonstationary nature. In recent years, research studies on learning in the presence of concept drift have been conducted through adaptive online learning approaches. Adaptive learning models update predictive models in nonstationary environments (NSE) where the concept is driven. Online learning is a sub-discipline of machine learning. It includes basic learning categories designed to provide sequential and continuous learning by entering data stream samples into the processing system. Online learning covers the inefficiencies of batch learning because online learners can quickly and efficiently update their models when new training data arrives. In addition, there are other advantages to online learning models, such as ease of use and understanding (Hoi et al., 2018). According to the necessity to make learning models efficient for real-time large-scale data stream processing and analysis, online learning is finding more and more interest and application. Moreover, the underlying concept in which one is predicting objects may change over time. That is why adaptive learning comes into the picture to handle concept drift (Giraud-Carrier, 2000). Adaptive models are advanced incremental learning approaches that provide adaptability to a variety of changes and no-stationarities by updating modeling (Gama et al., 2014). Incremental online learning can instantly track and adapt to different kinds of concept drift. Among the recent contributions to online learning, detecting conceptual deviations has attracted researchers to develop solutions for dealing with conceptual deviations and changes.

Adaptive learning in an NSE is considered from two major viewpoints: active and passive approaches. The approach is passive when continuous learning comes without knowing the change of concept (Salazar et al., 2013; Brzezinski & Stefanowski, 2014). Compared to passive methods, active methods use triggering mechanisms such as change detection tests (CDT) (Alippi, 2014). The adaptability of the models to the NSE environment in the triggering mechanism is done if the change signal is issued from a CDT test. Active online approaches benefit from explicit detection of the change in concept drift (CD), using the concept drift detection method (CDD) to activate an adaptation mechanism. The concept of the data stream is monitored with indicators to see the occurrence of concept drift (Jaworski et al., 2017; Liu et al., 2018). It is highly expected that the learning concept algorithm maintains the quality of learning while the monitoring system performs the expected functions. There has been a considerable exertion exploring methods for detecting concept drift. During the last decade, several methods of CDD have been developed .

Active online learning that needs efficient change or drift techniques can be used in pricing strategy or energy demand analysis. This task envisions sudden changes in the concept. The focus is on nonstationary adaptive and active monitoring to respond to concept drift better. Existing active drift detection algorithms mainly control labeling ratio under a fixed budget and compare it with parameters from the known distribution; Poisson and Binomial. Evolutionary characteristics of an NSE usually lead to lower performance of supervised learning. The accuracy decreases when the distribution of concepts changes. In other words, the state of occurrence of the concept drift alters.

Online learning and model adaptability were viewed and addressed using two main approaches. Several instances are continuously taught to update the model in both online and batch learning. (Wang et al., 2003). Contrary to batch learning, the cross-validation method used in stream data is not suitable for two reasons: an unlimited number of samples and evolving distributions of the samples (Gama, 2010). Standard CDD monitors any form of error rate for a predictive model or model validation. Model validation measurement accuracy of a hypothesis is the probability of accurately classifying random instances. Hereafter, accuracy is used as an index of learning performance and a practical tool for improving natural large data resources algorithms. The drift detection method used in our proposed online framework provides active drift detection.

The major inefficiencies of the precedent methods are the need to make assumptions about concepts or the conditional probability distribution of stream data. Besides, they are usually devoted to a known and constant distributional parameter. The method proposed in this paper is not based on the assumption that the learning rate distribution is known with a stationary parameter.

The underlying skeleton of this paper is carefully designed into five main sections: section 2 presents a concise outline of concept drift techniques and adaptive online methods. In the third section of this article, the proposed CCDD methods in adaptive online learning will be dealt with in detail. Section 4 provides results and discussions. Our concluding remarks and further research ideas are provided in the final section.

2. Literature review

Online methods are one of the most recently developed models of machine learning models, which, unlike batch models, could update the online best predictive model with the arrival of samples. Online learning can reduce the deficiency of batch learning therein; the predictive model can be updated instantly for any new arriving stream instances. Accordingly, online learning algorithms are, to a greater extent, competent and are used for large-scale machine learning tasks in real-world data analytics applications of their scalability feature. Big data streams are extensive and arrive at a high velocity. Therefore, a more realistic assumption, which also defines incremental learning, is just to accept that, stream at time t, S_t, is only available for learning or evaluation when first presented to the algorithm, which is additionally characterized as one-pass learning (Polikar et al., 2001; Kuncheva, 2004; Bifet et al., 2009a). Secondly, the majority of concept drift detection techniques suppose that the predictions made by the supervised learning will be verified with the tags for S_t arriving along with the subsequent training dataset S_t. This setting enables the model to check a loss value at every time step and is well-known as the test-then-train approach. The previous dataset is evaluated before training with the following dataset (Bifet, 2009; Bifet et al., 2009a).

Concept drift is one of the most popular problems in Stream Analytics, especially in online or incremental techniques. As time goes on, the performance of a learning model may become increasingly unreliable. This problem is known as concept drift and is a common challenge in a real-world scenario. When concept drifts are present, examples from the new distributions are of the highest importance to the learner, as they must be utilized to maintain competitiveness. Detecting change in probability distributions could be a critical issue in-stream analytics because of the advance in applications managed by such data. Any algorithm that does not make the required adjustments to changes in data distribution will ultimately fail to produce satisfactory generalization on future data if such data do not come from the distribution on which the algorithm was initially trained. Machine learning models supporting the active online approach expect to detect concept drift based on a change detection mechanism by explicit methods. In an active concept drift detection method, the model is continuously and sequentially updated upon receiving the signal of the concept drift. Since the concept drift is detected, the mining or learning model will perform corresponding action based on the detected drift, updating the learning model with the recent samples executing window management, or developing a new learning model to learn the incoming data from scratch (Žliobaitė et al., 2013).

Recently, some research focused on active drift detection. Active drift detection's state of the art includes a drift detector proposed in active drift detection research using the Restricted Boltzmann Machine (RBM) (Jaworski et al., 2017) and active learning approaches proposed using representative examples of the new concept and retraining of predictive models (Krawczyk et al., 2018).

Explicit methods implement concept drift tracker and triggering threshold to signal adaptation module to handle detected drifts (Gama et al., 2014). Concept drift models in change alarm systems give the system the ability and feature of interactive management to adapt to drift. In explicit approaches, concept drift management is conducted by monitoring the performance of flow data learning models, which in most studies have been either classification models (Gonçalves Jr et al., 2014; Khamassi & Sayed-Mouchaweh, 2014) or the probability distribution (Cieslak & Chawla, 2009; Lichtenwalter & Chawla, 2009; Ditzler & Polikar, 2011). However, it is a very complex task to implement concept drift detection approaches in a way that identifies the types of possible drifts. (Gama et al., 2014).

Adaptation in the general mechanism of adaptive models in concept drift is made with two main approaches, active and passive (Elwell & Polikar, 2011; (Alippi & Roveri, 2008). There are distinctive control strategies in CD approaches to issue drift compliance commands, in which the use of performance indicators is practical.

When the concept change reaches the alert level, three steps are required to handle concept drift:

(1) *Monitoring step*: in a nonstationary context, informed methods are applied to monitor the location, speed, and severity of evolution (Ikonomovska et al., 2009). These methods are popular as an indicator for triggering mechanisms in the adaptive learning cycle, where facilities adapt tracing indices such as learners' performance, the estimate of distribution, or the learner's structure and parameters. This step is applied in the explicit method, whereas implicit approaches ignore it and the following steps to assess the quality metric of the learning process. The main

challenges are (1) existing a sequential stream of data rather than constant batches of identically independent distributed instances, (2) decision support models continuously improve rather than being static with time, and (3) the data is produced by nonstationary probability distributions instead of a stationary one.

(2) *Updating step*: updating the learner model to adapt drift in explicit methods with concept drift detectors can be either re-learning from scratch or updating it using recent data after triggering the signal. Deciding on a window size strategy and forgetting strategy is a challenge in this step (Sobhani & Beigy, 2011). Some studies have a solution in window size optimization by enlarging the size dynamically when the drift is detected (Baena-Garcıa et al., 2006). Ikonomovska et al. (2009) progressively adapted learners using multiple windows with different sizes to handle concept drift. The blind approach implicitly makes necessary adaptation to learning processors by arriving at the ongoing concept at regular intervals (Kolter & Maloof, 2007).

(3) *Diagnosis step*: the changes in concepts and structure is interpreted in this step.

In the explicit approach, the following performance indicators of the learning model are used to monitor the concept drift (Klinkenberg & Renz, 1998):

- Statistical modeling performance indicators
- Model properties (complexity or dependencies)
- Descriptive characteristics or data properties

The next section is devoted to providing a brief review of the most famous drift detection methods used in the result section as competing detectors.

3. Online adaptive learning with conjugate concept drift detection

In online modeling of an environment in the NSE context, it is important to detect the significant concept drift through performance monitoring continuously. The triggering mechanism in the explicit method for handling concept drift improved the generic model developed by Gama et al. (2014) and Pastorello et al. (2017). The current paper seeks to maximize the prediction accuracy as a supervised performance to develop a non-defective model for the production process over time with minimum variance. The proposed concept drift detection method makes use of a trigger mechanism interactively. The concept drift of the data stream is detected when the signal is issued by the control system (Baena-Garcıa et al., 2006; Castillo & Gama, 2006, Gonçalves Jr et al., 2014; Khamassi & Sayed-Mouchaweh, 2014). When the concept drifts experience the alert level, the evaluation methodology in drift environments starts. The generic way to conduct concept drift handling technique is as following pseudo-components:

- Collecting the data set for training and curating
- Training the analytical model
- Monitoring the model continuously
- Checking if the alarm signal component calls for retraining
- Retraining the model

A tradeoff exists between the prediction validation metrics and the efforts for computational complexity reduction needed to update and estimate the proper amount of data used for refining the learning model after CDD ordered a reaction. The optimal tradeoff guarantees satisfactory performance of the proposed training strategy when an ML (Machin Learning) model is updated.

According to the abovementioned generic way to monitor concept drift, the proposed module in this paper works in three phases, as shown in Figure 1:

1) Learning
2) Concept drift detection in streaming data
3) Adaptation

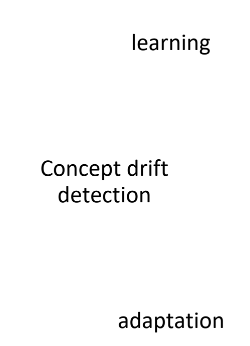

Figure 1: The proposed online adaptive framework

This paper is about active online learning in dynamic environments, for which new active learning strategies are proposed in charge of drift detectors. The main idea is to focus on the performance of any learning modeling, especially incremental ones tracked by their probability distribution. Since the evolution characteristic of NSE neither guarantee the type of distribution nor the probability characteristics (such as the location and the scale parameters) remain unchanged, the evolution of the pdf parameters is considered. In what follows, the above three phases are discussed in detail.

3.1. Learning phase

This paper focuses on the classification under supervised learning for incremental/online algorithms. This learning algorithm is expected to be triggered several times sequentially on different chunks of a dataset to use out-of-core or online learning. Online learning is particularly applicable in case the data streams are ambitiously large scale to fit in memory at once. Naive Bayes (NB) classifiers support sample weighting. NB provides fewer parameter estimates compared to other rival models. Accordingly, the problem of overfitting inclined even to moderating a small sample size. The traditional feature pre-selection eliminates correlated attributes; Hence, the assumption of independence will rarely be valid for a subset of variables (Kuncheva, 2006). This algorithm learns efficiently from a few training examples (Salperwyck & Lemaire, 2011). Naive Bayes classifiers have performed exceptionally well in many complex real-world situations (Zhang, 2005). An analytical study compared the performance of the NB algorithm with that of the Random Forest and boosting classification algorithms. The result of the analysis showed a remarkable contrast of NB (Caruana & Niculescu-Mizil, 2006). Therefore, in selecting the learning model, the Gaussian Naive Bayes (GNB) model is employed to use the adaptability of this model in NSE. Thus, the GNB is chosen in the current research to model the adaptation better by using the two features of Bayesian estimation and updating the mean and variance parameters alongside using the prequential evaluation method used in concept drift models.

Gaussian NB is a binary classifier. It can perform online updates to model parameters via partial_fit. In this approach, the likelihood of the features shown in Equation (1) is assumed to be Gaussian:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}}\exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right), \qquad (1)$$

where the parameters σ_y and μ_y are the estimates of the mean and variance using the maximum likelihood estimator (MLE) with the following log-likelihood function

$$L(\mu;\sigma^2;x_i \mid y) = log\prod_k P(x_i^{(k)} \mid y) = \sum_k \log P(x_i^{(k)} \mid y). \tag{2}$$

GNB provides partial fit by updating the mean and variance of sample for the Naıve Bayes classifier through successive variance sampling estimation using the following pairwise algorithm based on two samples $\{x_i\}_{i=1}^{m}$, and $\{x_i\}_{i=m+1}^{m+n}$ (Chan et al., 1982):

$$S_{1,m+n} = S_{1,m} + S_{m+1,m+n} + \frac{m}{n(m+n)}((m+n)/m\, T_{(1,m)} - T_{(1,n+m)})^2, \tag{3}$$

where; $T_{(1,m)} = \sum_{i=1}^{m} x_i$, $T_{(\mathrm{m+1,n})} = \sum_{i=m+1}^{n} x_i$, $T_{(1,n+m)} = T_{(1,m)} + T_{(1,n+m)}$, and $S_{(1,n+m)} = S_{(1,m)} + S_{(m+1,n+m)} + \frac{m}{n(m+n)}\left(\frac{n}{m}T_{(1,m)} - T_{m+1,m+n}\right)$.

The above predictive sequential (or prequential method) is used since it is recommended to gain superior adaptation performance in stream settings contrary to its classical competitors, namely, hold-out evaluation (Chan et al., 1982). Each arrival sample tests the Naıve Bayes classifier through this method and then trains the GNB model or partial fitting. For prequential evaluation, the instances are first used to test and then train and return the label predictions for each test instance and the associated running time (Gama et al., 2013).

$$P_\alpha(i) = \frac{\sum_{k=1}^{i} \alpha^{i-k} L(y_k,\hat{y}_k)}{\sum_{k=1}^{i} \alpha^{i-k}}, \tag{4}$$

where α $(0 < \alpha \leq 1)$ is a constant determining the forgetting factor of the summation of the sample window, which should be close to 1 at time i, over a sliding window of size w and $k = i - w + 1$. The prequential error is computed based on an accumulated sum of a loss function between the prediction and observed values if a fading factor doesn't consider $P_\alpha(i) = \sum_{i=1}^{n} L(y_k,\hat{y}_k)$.

3.2. Explicit triggering phase using concept drift detection

This phase is a core module of our proposed approach designed to detect concept drift comparable to the mentioned method stated in Section 2. The quality of the learning algorithm is said to have an accuracy of $1 - \varepsilon$ with reliability of $1 - \delta$. Inspired by the probably approximately correct (PAC) approach, the rate of error-dependent learning track to check the stationarity of probability distribution. Therefore, increasing the error of the learning model is useful when it is statistically significant as an effective indicator of change in probability distribution (Khamassi et al., 2015). Drift detection methods, especially the DDM family, which monitors the error rate of the learning model, assume an identical distribution of error rate or the quality metric.

In contrast to the EDDM method (Baena-García et 2004), and other related techniques based on the error of classification model, the presented drift detection method (CCDD) is a true counted-based drift detection method. Referring to the volatile and nonstationary nature of concepts and data, the assumption that the distribution of this assumption is unchanged is violated; therefore, the Bayesian approach to statistical inference is implemented when the predictions of the sequential performance of incremental learning are evaluated. Consequently, we present conjugate estimation in the triggering phase to improve the DDM model under unknown, uncertain parameter conditions and the need for continuous parameter estimation updating instead of the classical estimation method in the DDM family. Distributional distance-based limits are introduced to provide a high-efficiency control chart concerning the proper threshold,. The abovementioned solution is implemented with three metric measures: HPDI, Hellinger, and Kullback Leibler (KL). These distances are used to compare the current count data with the reference sample.

3.2.1. Concept drift detection with Beta-Bernoulli conjugate distribution

In the first scenario, by arriving stream at time t, the number of true predicted according to the actual value x_i is considered as the quality performance of the learning model. The processing models are fitted online when the data stream arrives; each time, the true count index of the model is monitored as the system's quality. As the number of correct labels follows a binomial distribution, estimating the current time rate according to the binomial probability distribution is possible. The future rate can also be estimated.

Let x_i be i.i.d. output data following the Binomial(m,p) distribution. Here the two possible outcomes are recorded by assessing the results in the form of the correct predictions (T) or the incorrect ones (F). In other words, true and false are the results of comparing the outcomes of the processed model with the actual outcome. The rate of correct predictions for the test data, which can be calculated easily by dividing the number of correct predictions by the number of total predictions, is used as a machine learning performance metric. It indicates that the processing model quality is calculated sequentially by arriving at the instances. The streams of assessed values have NSE characteristics since the environment has such property. To deal with NSE characteristics, we suggest estimating the parameter sequentially and releasing the constraint, assuming that a constant value is required for the assumed parameter. Therefore, estimating the probability distribution parameter of the accuracy rate in each stream, p_{acc}, is assessed using the Bayesian approach. To estimate p_{acc}, a Bayesian analyst would put a prior distribution on p_{acc}. The prior distribution would be p_{acc}~B(α,β). Given a sample $x_1, x_2, \ldots, x_n$, the Beta conjugate estimation assumes a Beta distribution with the parameters given in Equation (5) as the posterior distribution for Binomial distribution, i.e.,

$$p_{acc} \,|x \sim Beta(\alpha + \textstyle\sum_{i=1}^{n} x_i, \beta + mn - \sum_{i=1}^{n} x_i), \tag{5}$$

where α and β are the first and second parameters of the beta distribution, respectively, n is the number of arriving instances, and m is the number of binomial trials for$(x\,|\,p)$.

Bayesians prefer to work with HPD (highest posterior density) credible sets $A = \{\theta{:}h(\theta\,|\,x_1, x_2, \ldots, x_n) \geq K(\alpha)\}$, where $K(\alpha)$ is the largest real number such that P(A) ≥ 1 − α. For a unimodal posterior, the set becomes an HPD credible interval (Turkkan & Pham-Gia, 1993). The monitoring design works with HPD interval in a way that whenever p_1 is within the interval [l; u] and p_2 is outside the interval, then

$$f(p_{acc\,i}\,|\,data) \geq f(p_{p_{acc\,i+1}}\,|\,data). \tag{6}$$

Figure 2 depicts the steps taken in the proposed CBBC_HPDI drift detection method. The pseudocode of CCB_HPDI is brought in Figure 3.

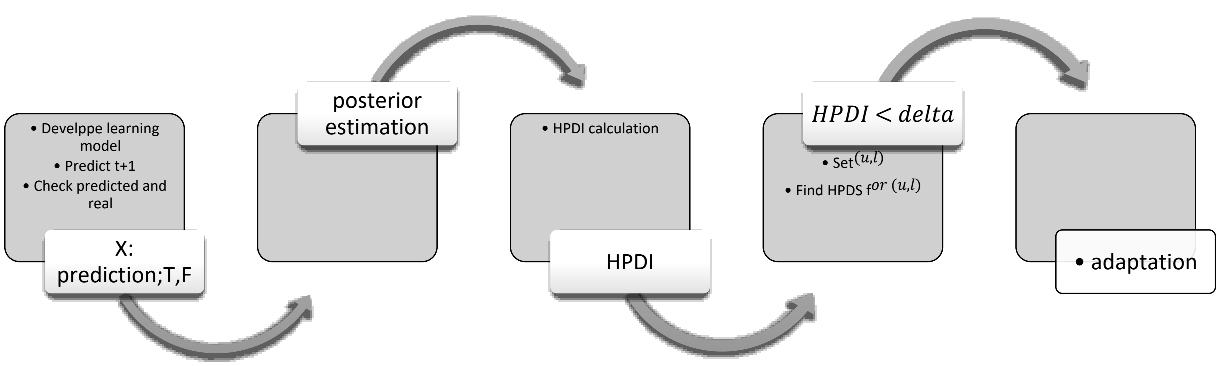

Figure 2: The CBBC_HPDI drift detection method

3.2.2. Concept drift detection with Gamma-Poisson conjugate distribution

The other practical solution for detection and prediction is based on conjugate analysis in which the classification accuracy is of rate type and can be fitted by a Poisson distribution. As the distribution parameter is unknown in NSE, we refer to Bayesian conjugate analysis to estimate the

Poisson parameter by a Gamma distribution so that any processing system in an NSE should maintain a level of expected quality over time. Therefore, the performance quality parameter of the model is tracked so that the change of the parameter detects drifts. In the following sections, the model quality parameter is estimated from the two Bayesian methods, and then the warning method is determined.

Algorithm1: CCB_HPDI
Input: Boolean; 1 if the prediction is incorrect; 0 otherwise (any learner, but here Gaussian naïve Bayes) Delta; w, alpha, l, u; Output1: predicted parameter of performance (λ) (with beta-Bernoulli conjugate) Ouput2: change detection

#testing by HPDI#
 	
hpdi = cdf(u) - cdf(l)
    # change_detected = True if hpdi < delta_ else False
    if change_detected:
      _reset_params()

Figure 3: The Pseudo-code of the proposed CCB_HPDI algorithm

3.2.2.1. The CCDG-H drift detection

In this method, it is also assumed that the data is entered into the processing system as a data stream. The accuracy of the learning model in the second scenario is considered a quality performance. The models are fitted online repeatedly; each time, the accuracy index of the model is monitored as the quality of the system. Assuming the accuracy rate to follow a Poisson distribution, estimating the current rate is possible based on the Poisson probability distribution. The future rate can also be estimated. Therefore, assuming that $x_i \sim Poisson\,(p)$, the Gamma distribution with the parameter $\alpha + \sum x_i/(n + 1/\beta)$ is used as the posterior distribution for Poisson. This means that the conjugate estimator of the rate, $\hat{p}_{acc}$, obtained by Bayesian inference is:

$$\hat{p}_{acc} = \alpha + \sum x_i/(n + 1/\beta). \tag{7}$$

Figure 4 shows how the proposed (CCDG-H) drift detection approach works.

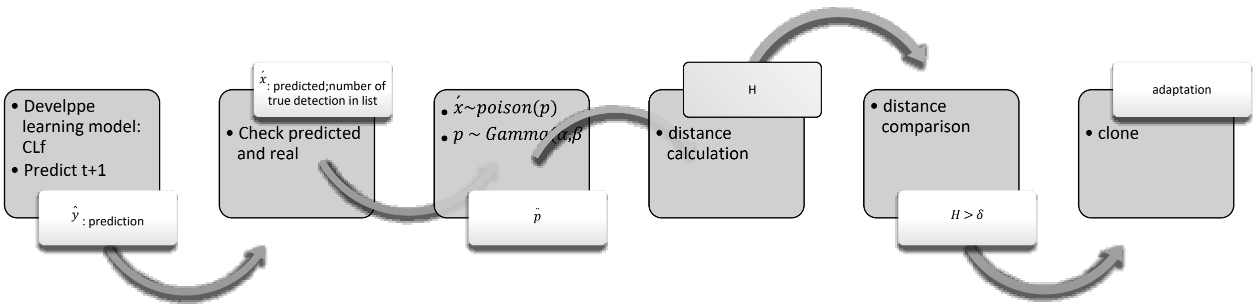

Figure 4: CCDG-H drift detection method

Regardless of the concept drift, the Hellinger distance is used to measure the similarity between the distribution of training data and test data in early incremental learning studies (Cieslak & Chawla, 2009). Hellinger distance quantifies the similarity between two probability distributions. Hellinger metric with an adaptable threshold is proposed to compare the incoming batch of data with reference one (Ditzler & Polikar, 2011). Some distance measures to compare the probability distribution of windows were implemented by Kifer et al. (2004). Since the Hellinger distance is symmetric against the Kullback-Leibler (KL) divergence criterion, it would be more appropriate.

The Hellinger distance $H(p_{acc\ i}, p_{acc\ j})$ is computed from the consecutive probability density $p_{acc\ i}$ and $p_{acc\ i+1}$, indicating the divergence difference. The Hellinger distance between two distributions, $p_{acc\ i}$ and $p_{acc\ j}$, has a general form given by

$$H(p_{acc\ i}, p_{acc\ j}) = \frac{1}{\sqrt{2}}\sqrt{\sum_{x \in X \cup Y}\left(\sqrt{P(p_{acc\ i})} - \sqrt{P(p_{acc\ j})}\right)^2}. \qquad (8)$$

Subsequently, the difference between the probability distributions of sequential data flows, which is our statistic in the study, is going to be compared with the threshold limit. The evaluation of these statistics with the threshold shows either the absence or the presence of concept drifts. Using this system, we have an alarming machine that, on the premise of having a larger value than the delta, triggers the rebuilding of the model. To reach a better perfomance, the threshold is usually already defined. For example, to achieve the accuracy of at least 70%, we set the default value of 0.02 for δ. The pseudocode of the presented Conjugate Concept drift Hellinger distance method (CCG –H) is shown in Figure 5.

Algorithm2-1: CCG –H
Input performance of learning and detection; accuracy (any learner, but here Gaussian naïve Bayes) Alpha1=1, beta1=0.5, $\delta = 0.02$ Number of instances in the window: default: w=30 Sliding window: w ed recent instances
#estimation: ***Output: predicted parameter of performance (with Gamma-Poisson conjugate)*** $p_{acc\,i}$ $p_{acc\,j}$
Calculating Hellinger distance: $D_H = \frac{1}{\sqrt{2}}\sqrt{\sum_{x \in X \cup Y}\left(\sqrt{P(p_{acc\,i})} - \sqrt{P(p_{acc\,j})}\right)^2}$
#detecting drift: testing by Hellinger distance# Change is detected if: $D_H > \delta$

Figure 5: The pseudocode of the CCG-H approach

3.2.2.2. The CCDG-KL drift detection

The KL divergence, introduced as relative entropy, computes the loss function for two probability distributions. The KL divergence of distribution $p_{acc\,j}$ from $p_{acc\,i}$ is a measurement of the loss function when $p_{acc\,j}$ is estimated to approximate $p_{acc\,i}$. It should be noticed that KL divergence, as one of the most common measures, is not a true distance metric. In change detection literature, the Kullback-Leibler divergence exhibits a high probability of change detection, faster detection rates, and few false positive alarms compared to Cosin distance (Sebastiao & Gama, 2007). The general Equation to calculate the KL divergence between two Gamma distributed random variables, $p_{acc\,i}$ and $p_{acc\,j}$ is given by

$$D_{KL}((p_{acc\,i})\|QP(p_{acc\,j}))\sum P(p_{acc\,i})log\left(\frac{P(p_{acc\,i})}{P(p_{accj})}\right). \quad (9)$$

With the KL distance, a threshold can be defined to monitor the system and see whether there is a significant change between the two distributions at two consecutive timestamps. The calculated distance value indicates the distance and deviation of the two parameters. This threshold would be user-defined with the desired accuracy. We set the threshold δ to 0.01 as a default value. The pseudocode of the proposed Kullback-Leibler (KL) distance drift detection method (CCG – KL) is shown in Figure 6.

Algorithm2-2: CCG – K L
Input performance of learning and detection; accuracy (any learner, but here Gaussian naïve Bayes) Alpha1=1, beta1=0.5, $\delta = 0.01$ Number of instances in window: w=30 Output: predicted parameter of performance (with Gamma-Poisson conjugate)
#estimation: ***Output***: predicted parameter of performance (with Gamma-Poisson conjugate) $\boldsymbol{p_{acc\ i}}$ $\boldsymbol{p_{acc\ j}}$
#testing by KL distance# $D_{KL}((p_{acc\ i})\|QP(p_{acc\ j}))\sum P(p_{acc\ i})log\left(\frac{P(p_{acc\ i})}{P(p_{accj})}\right)$ Change is detected if: $D_{KL} > \delta$

Figure 6: The pseudocode of the proposed CCG_KL approach

3.3. Adaptation phase

In the active approach mechanism based on change detection, adaptation to change takes place through re-learning. The re-learning method of this paper is a complete re-learning or clone approach that re-teaches the supervisory learning model, here GNB, in the learning process (Zhang, 2004).

4. Experimental results and analysis

The set of ideas proposed in this paper has been examined for the Electricity dataset, which is popular for adaptive models when concept drift is present. It has been employed in more than 50 concept-drift scientific tests. The dataset includes 45,312 Australian New South Wales Electricity data registered every half an hour with six input variables over two years from May 1996 to December 1998.) The rates are not stable in this market and are usually impressed by the demand and supply changes. The class tag identifies the rate change (either up or down) related to a moving average of the last 24 hours is identified by the class tag (label). The electricity market rates are set every minute by matching the current demand for electricity with the cheapest blend of electricity from all power stations, corresponding to rate schedules published and updated by each power station. Rate schedules set the rate for different levels of electricity production. Market demand and supply determine what the electricity rates should be. Factors including time of the day, season, weather, etc., are major factors seen as a drift cause on market demand. This dataset has a critical property that merits consideration when assessing concept drift adaptation. Assume a simple predictor is employed to predict the following tag to be the same as the current one. That is, if the rate goes up now, it would predict that it will be the same next time (the rate will go up again in the next step). In case the data is distributed apart, such a predictor would reach 51% of accuracy. However, it will give a much higher 85% of accuracy if we implement this Naïve method on the Electricity dataset. This results from the absence of independence between labels. This dataset is subject to concept drift due to seasonality, unexpected events, and changing consumption habits. Three of the five covariates have a dominant effect on concept drift. To implement scenario 1, the code related to Algorithm 1 uses the following input values. The number of window samples: w=1000, alpha_=0.05, l=.60, u=0.75, and Delta=0.01.

4.1. Analysis of the results

The experimental study compares popular classification stream mining having common drift detection with the proposed conjugate concept drift detection (CCD) concerning change monitoring distance method, including CCG_Hellinger, CCG_KL, and CCB_HPDI. In this section, a discussion is carried out on the performance of the competitive method. The interpretation for analysis in a method comparison is conducted by evaluating CCD against that of DDM, EDDM, ADWIN, and HDDMA families.

Initially, based on the comparison of the accuracy index, the methods are evaluated under NSE. As shown in Figure 7, the CCG_H method has superior performance to the other competitors. Besides, Table 1 shows the experimental results of the experiments obtained using the drift detectors. It can be observed in Table 4 that, in terms of drift detection capability, one of the considerable criteria is the count of drifts included against the accuracy of the processing system. Based on the extracted values, it is inferred that while approach 1 has a lower number of drifts than the ADWIN model, the average quality index with a value of 0.77 has a more favorable advantage over competing models. One of the notable successes of the proposed approach with the Hellinger method is that this method is able to increase the accuracy criterion as a quality index of statistical modeling and, in other words, the non-defective index by 4% as a tool to identify the concept drift in the model adaptation approach. Another criterion considered in data processing is the variance of the model accuracy computed for each batch, including 1,000 samples. As the variation of quality metrics plays an essential role in quality improvement, sample variation is considered an additional parameter for comparing drift detectors. In the Hellinger method, the variance of the model quality index is 0.0035, which has the minimum variance among the drift detector as competitors.

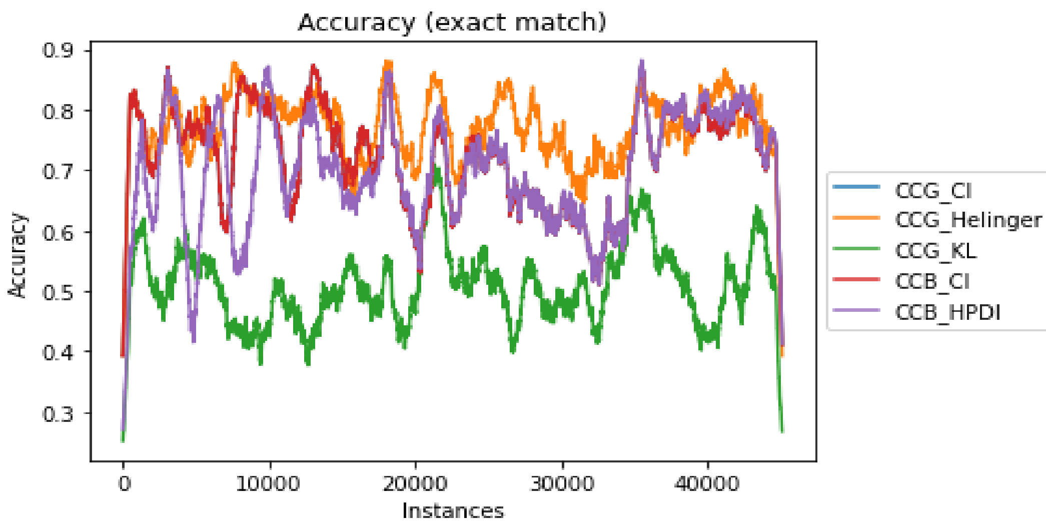

Figure 7: Accuracy comparison

Table 1: Mean accuracy and variance within the window of 1,000 samples

Drift detector	**Drift detection**	**Mean acc**	**Variance acc**
CCG_Hellinger	73	**0.7729**	**0.0035**
CCG_KL	1456	0.5074	0.0043
CCB_HPDI	4165	0.7071	0.0086
Adwin	139	0.7496	0.0038
Page-Hinkley	8	0.7342	0.0078
DDM	360	0.7042	0.0076
EDDM	744	0.5865	0.0042
HDDM_A	360	0.6802	0.0091
HDDM_W	250	0.7224	0.0062
GaussianNB	-	0.7289	0.0074

In Table 2, the results of the time and memory consumption of the models in megabytes are compared. These indicators are considered in terms of processing large volume data. Based on the results, the CCG_Hellinger approach requires about 50% more total processing time than Adwin and Page models. The maximum memory usage for CCB_HPDI, Adwin, and Page-Hinkley are the same, where CCG_Hellinger needs 10% more. The general conclusion of the results is that, in situations where the average accuracy is continuous with minimal variance in unstable conditions or a nonstationary environment containing concept drift, where processing time is relatively important, the Hellinger model is desirable. Figure 8 shows an accuracy comparison between CCG_H and PH, DDM, HDDM, and EDDM methods.

Table 2: The total process time and maximum usage

Drift detector	Total process time	Maximum memory usage
CCG_Hellinger	54.96	384.78
CCG_KL	23.06	368.87
CCB_HPDI	33.93	347.54
Adwin	23.37	347.54
Page-Hinkley	22.625	347.54
DDM	22.382	368.87
EDDM	22.586	368.87
HDDM_A	22.836	368.87
HDDM_W	22.8969	368.87

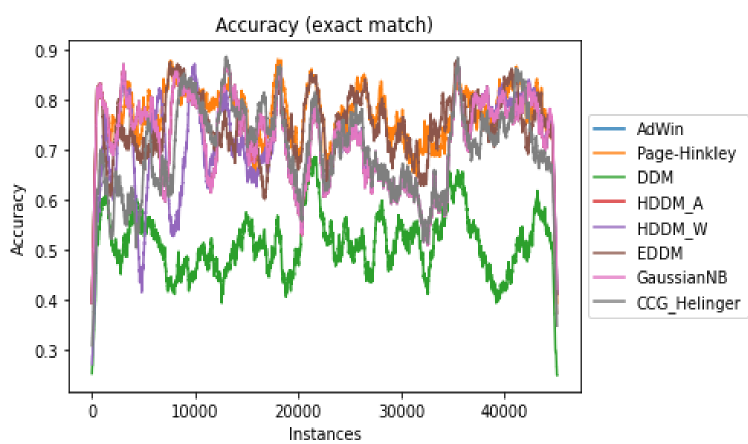

Figure 8: Accuracy comparison between CCG_H and PH, DDM, HDDM, EDDM

5. Conclusion and recommendations for future research

Based on DDM, a conjugate concept drift detector (CCG_H) was proposed in this paper as an active concept drift detector for volatile stream settings. In contrast to CCD, which needs parameter information about the distribution of events of the learning model, the proposed method is not based on fixed distributional parameters when concepts are evolutionary. It works under nonstationary environments and provides adaptivity. The adaptivity comes from taking advantage of three approaches: (1) an online active learning method with partially updating learning parameters, (2) a conjugate estimation of the learning model, and (3) an explicit triggering based on an updated estimation compared with Hellinger distance. This approach might as well be used as a separate module inspecting the system to detect sudden changes in the probability distributions or, as we mentioned, a concept drift detector.

In the current paper, according to its application in binary labeled variables, the aforementioned adaptive system was confirmed because of the better performance and fitness of the Gaussian Naive Bayes (GNB) learning model with binary labels. To evaluate the proposed method against DDM, RDDM, ADWIN, and PH, the Gaussian Naïve Bayes approach was deployed as the base learners. Experimental computation on a popular Electricity dataset with 45,312 records showed that the Hellinger model works the best in a nonstationary environment containing concept drift, where the highest average accuracy with minimal variance is desired and the processing time is relatively important. It was also shown that the tool could deal with both sudden and gradual concept drifts.

The proposed approach can be applied in reliability analysis, where the mean time between failures (MTBF) and distributional reliability inference is considered. The latter approach would consider the quality of the processing system in terms of time and degradation modeling to be aware of learning performance. Distribution estimation supplies tools to have a maintenance approach to predict learning failure. The metric time between detection and time between false alarms would be monitored to predict future behavior. So, reliability and survival analysis and a rare event control chart might be helpful.

Reference

Alippi, C. (2014). Intelligence for embedded systems. Berlin, Germany, Springer-Verlag.

Alippi, C., Roveri, M. (2008). "Just-in-time adaptive classifiers—Part II: Designing the classifier." IEEE Transactions on Neural Networks **19**(12): 2053-2064.

Bach, S. H., Maloof, M. A. (2008). Paired learners for concept drift. 2008 Eighth IEEE International Conference on Data Mining, IEEE.

Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo-Merino, R., Bifet, A., Gavald, R., Morales-Bueno, R. (2006). Early drift detection method. Fourth International Workshop on Knowledge Discovery from Data Streams, https://www.researchgate.net/publication/245999704_Early_Drift_Detection_Method

Bifet, A. (2009). "Adaptive learning and mining for data streams and frequent patterns." ACM SIGKDD Explorations Newsletter **11**(1): 55-56.

Bifet, A., Gavalda, R. (2007). Learning from time-changing data with adaptive windowing. Proceedings of the 2007 SIAM international conference on data mining, SIAM. https://doi.org/10.1137/1.9781611972771.42

Bifet, A., Holmes, G., Pfahringer, B., Gavald`, R. (2009a). Improving adaptive bagging methods for evolving data streams. Asian conference on machine learning, Springer. https://researchcommons.waikato.ac.nz/bitstream/handle/10289/3646/HatBag.pdf?sequence=1

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., Gavaldà, R. (2009b). New ensemble methods for evolving data streams. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM. https://www.cs.upc.edu/~abifet/fp391-bifet.pdf

Brzezinski, D., Stefanowski, J. (2014). "Reacting to different types of concept drift: The accuracy updated ensemble algorithm." IEEE Transactions on Neural Networks and Learning Systems **25**(1): 81-94.

Cabral, D. R. L. (2017). Statistical tests and detection of concept drifts in data streams. Portuguese, Universidade Federal de Pernambuco.

Caruana, R., Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In: Fürnkranz J., Scheffer T., Spiliopoulou M. (eds) Knowledge Discovery in Databases: PKDD 2006. PKDD 2006. Lecture Notes in Computer Science, vol 4213. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11871637_11

Castillo, G., Gama, J. (2006). An adaptive prequential learning framework for Bayesian network classifiers. n: Fürnkranz J., Scheffer T., Spiliopoulou M. (eds) Knowledge Discovery in Databases: PKDD 2006. PKDD 2006. Lecture Notes in Computer Science, vol 4213. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11871637_11

Chan, T. F., Golub, G. H., LeVeque, R. J. (1982). Updating formulae and a pairwise algorithm for computing sample variances. In: Caussinus H., Ettinger P., Tomassone R. (eds) COMPSTAT 1982 5th Symposium held at Toulouse 1982. Physica, Heidelberg. https://doi.org/10.1007/978-3-642-51461-6_3

Cieslak, D. A., Chawla, N. V. (2009). "A framework for monitoring classifiers' performance: when and why failure occurs?" Knowledge and Information Systems **18**(1): 83-108.

de Barros, R. S. M. (2017). Advances in data stream mining with concept drift. Brazil, Universidade´ Federal de Pernambuco.

de Barros, R. S. M., Hidalgo, J. I. G., de Lima Cabral, D. R. (2018). "Wilcoxon rank sum test drift detector." Neurocomputing **275**: 1954-1963.

de Lima Cabral, D. R. and R. S. M. de Barros (2018). "Concept drift detection based on Fisher's Exact test." Information Sciences **442**: 220-234.

Ditzler, G., Polikar, R. (2011). Hellinger distance based drift detection for nonstationary environments. 2011 IEEE symposium on computational intelligence in dynamic and uncertain environments (CIDUE), IEEE.

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

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

Gama, J. (2010). Knowledge discovery from data streams, CRC Press.

Gama, J., Medas, P., Castillo, G., Pereira Rodrigues, P. (2004). Learning with drift detection. In: Bazzan A.L.C., Labidi S. (eds) Advances in Artificial Intelligence – SBIA 2004. SBIA 2004. Lecture Notes in Computer Science, vol 3171. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-28645-5_29

Gama, J., Sebastião, R., Pereira Rodrigues, P. (2013). "On evaluating stream learning algorithms." Machine Learning **90**(3): 317-346.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A. (2014). "A survey on concept drift adaptation." ACM Computing Surveys (CSUR) **46**(4): 1-37.

Giraud-Carrier, C. (2000). "A note on the utility of incremental learning." Ai Communications **13**(4): 215-223.

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

Hoi, S. C., et al. (2018). "Online learning: A comprehensive survey." arXiv preprint arXiv:1802.02871.

Huang, D. T. J., Koh, Y. S., Dobbie, G., Pears, R. (2014). Detecting volatility shift in data streams. 2014 IEEE International Conference on Data Mining, IEEE.

Ikonomovska, E., Gama, J., Sebastião, R., Gjorgjevik, D. (2009). Regression trees from data streams with drift detection. In: Gama J., Costa V.S., Jorge A.M., Brazdil P.B. (eds) Discovery Science. DS 2009. Lecture Notes in Computer Science, vol 5808. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04747-3_12

Jaworski, M., Duda, P., Rutkowski, L. (2017). On applying the restricted Boltzmann machine to active concept drift detection. 2017 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE.

Khamassi, I., Sayed-Mouchaweh, M. (2014). Drift detection and monitoring in nonstationary environments. 2014 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), IEEE.

Klinkenberg, R., Renz, I. (1998). "Adaptive information filtering: Learning in the presence of concept drifts." Learning for Text Categorization: 33-40.

Kolter, J. Z., Maloof, M. A. (2007). "Dynamic weighted majority: An ensemble method for drifting concepts." Journal of Machine Learning Research **8**: 2755-2790.

Krawczyk, B., Pfahringer, B., Woźniak, M. (2018). Combining active learning with concept drift detection for data stream mining. 2018 IEEE International Conference on Big Data (Big Data), IEEE.

Kuncheva, L. I. (2004). Classifier ensembles for changing environments. In: Roli F., Kittler J., Windeatt T. (eds) Multiple Classifier Systems. MCS 2004. Lecture Notes in Computer Science, vol 3077. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-25966-4_1

Kuncheva, L. I. (2006). "On the optimality of Naive Bayes with dependent binary features." Pattern Recognition Letters **27**(7): 830-837.

Lichtenwalter, R. N., Chawla, N. V. (2009). Adaptive methods for classification in arbitrarily imbalanced and drifting data streams. In: Theeramunkong T. et al. (eds) New Frontiers in Applied Data Mining. PAKDD 2009. Lecture Notes in Computer Science, vol 5669. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-14640-4_5

Liu, A., Lu, J., Liu, F., Zhang, G. (2018). "Accumulating regional density dissimilarity for concept drift detection in data streams." Pattern Recognition **76**: 256-272.

Nishida, K., Yamauchi, K. (2007). Detecting concept drift using statistical testing. In: Corruble V., Takeda M., Suzuki E. (eds) Discovery Science. DS 2007. Lecture Notes in Computer Science, vol 4755. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-75488-6_27

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

Pastorello, G., Gunter, D., Chu, H., Christianson, D., Trotta, C., Canfora, E., Faybishenko, B., Cheah, Y.-W., Beekwilder, N., Chan, S., Dengel, S., Keenan, T., O'Brien, F., Elbashandy, A., Poindexter, C., Humphrey, M., Papale, D., Agarwal, D. (2017). Hunting data rogues at scale: Data quality control for observational data in research infrastructures. 2017 IEEE 13th International Conference on e-Science (e-Science), IEEE.

Pears, R., Sakthithasan, S., Koh, Y.S. (2014). "Detecting concept change in dynamic data streams: A sequential approach based on reservoir sampling." Machine Learning **97**(3): 259-293.

Pesaranghader, A., Viktor, H. L. (2016). Fast Hoeffding drift detection method for evolving data streams. In: Frasconi P., Landwehr N., Manco G., Vreeken J. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2016. Lecture Notes in Computer Science, vol 9852. Springer, Cham. https://doi.org/10.1007/978-3-319-46227-1_7

Polikar, R., Upda, L., Upda, S. S., Honavar, V. (2001). "Learn++: An incremental learning algorithm for supervised neural networks." IEEE Transactions on Systems, Man, and Cybernetics, part C (applications and reviews) **31**(4): 497-508.

Ross, G. J., Adams, N. M., Tasoulis, D. K., Hand, D. J. (2012). "Exponentially weighted moving average charts for detecting concept drift." Pattern Recognition Letters **33**(2): 191-198.

Salazar, D. S. P., Adeodato, P. J. L., Arnaud, A. L. (2013). "Continuous dynamical combination of short and long-term forecasts for nonstationary time series." IEEE Transactions on Neural Networks and Learning Systems **25**(1): 241-246.

Salperwyck, C., Lemaire, V. (2011). Learning with few examples: An empirical study on leading classifiers. The 2011 international joint conference on neural networks, IEEE.

Sebastiao, R., Gama, j. (2007). Change detection in learning histograms from data streams. In: Neves J., Santos M.F., Machado J.M. (eds) Progress in Artificial Intelligence. EPIA 2007. Lecture Notes in Computer Science, vol 4874. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-77002-2_10

Sobhani, P., Beigy, H. (2011). New drift detection method for data streams. In: Bouchachia A. (eds) Adaptive and Intelligent Systems. ICAIS 2011. Lecture Notes in Computer Science, vol 6943. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-23857-4_12

Turkkan, N., Pham-Gia, T. (1993). "Computation of the highest posterior density interval in Bayesian analysis." Journal of Statistical Computation and Simulation **44**(3-4): 243-250.

Wang, H., Fan, W., Yu, P. S., Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, http://hanj.cs.illinois.edu/pdf/kdd03_emsemble.pdf.

Zhang, H. (2004). "The optimality of naive Bayes." AA **1**(2): 3.

Zhang, H. (2005). "Exploring conditions for the optimality of naive Bayes." International Journal of Pattern Recognition and Artificial Intelligence **19**(02): 183-198.

Žliobaitė, I., Bifet, A., Pfahringer, B., Holmes, G. (2013). "Active learning with drifting streaming data." IEEE Transactions on Neural Networks and Learning Systems **25**(1): 27-39.