IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 1, Ver. II (Jan – Feb. 2015), PP 20-26
www.iosrjournals.org

A Review on Concept Drift

Yamini Kadwe[1], Vaishali Suryawanshi[2]

[1,2] *(Department of IT, M.I.T. College Of Engineering, Pune, India)*

Abstract: *The concept changes in continuously evolving data streams are termed as concept drifts. It is required to address the problems caused due to concept drift and adapt according to the concept changes. This can be achieved by designing supervised or unsupervised techniques in such a way, that concept changes are considered, and useful knowledge is extracted. This paper discusses various techniques to manage concept drifts. The synthetic and real datasets with different concept drifts and the applications are discussed.*

Keywords: *Drift detectors, ensemble classifiers, data stream mining, and bagging.*

I. Introduction

Today, is a world of advanced technologies, every field is automated. Due to advances in technology, plenty of data is generated every second. Examples of such applications include network monitoring, web mining, sensor networks, telecommunications data management, and financial applications [14]. The data needs to be gathered and processed, to extract unknown, useful and interesting knowledge. But it is impossible to manually extract that knowledge due to the volume and speed of the data gathered.

Concept drift occurs when the concept about which data is being collected shifts from time to time after a minimum stability period. This problem of concept drift needs to be considered to mine data with acceptable accuracy level. Some examples of concept drift include spam detection, financial fraud detection, climate change prediction, customer preferences for online shopping.

This paper is organized as follows, Section II gives the overview of concept drift, involving problem, need to adapt concept drift and types of concept drift. Section III explains various methods of detecting concept drift. Section IV discusses about statistical tests for concept drift. In Section V classifiers for dealing with concept drifts are discussed. Section VI gives a summary on possible datasets based on the type of drifts present and Section VII summarizes consideration of concept drift in various real-world applications.

II. Overview

Problem of Concept Drift:

There has been increased importance of concept drift in machine learning as well as data mining tasks. Today, data is organized in the form of data streams rather than static databases. Also the concepts and data distributions ought to change over a long period of time.

Need for Concept drift adaptation:

In dynamically changing or non-stationary environments, the data distribution can change over time yielding the phenomenon of concept drift[4]. The concept drifts can be quickly adapted by storing concept descriptions, so that they can be re-examined and reused later. Hence, adaptive learning is required to deal with data in non-stationary environments. When concept drift is detected, the current model needs to be updated to maintain accuracy.

Types of Concept drift:

Depending on the relation between the input data and target variable, concept change take different forms. Concept drift between time point t0 and time point t1 can be defined as-

$$\exists X : p_{t0}(X, y) \neq p_{t1}(X, y) \qquad (1)$$

where p_{t0} denotes the joint distribution at time t0 between the set of input variables X and the target variable y. Kelly et al. presented the three ways in which concept drift may occur [3]:

- prior probabilities of classes, p(y) may change over time
- class-conditional probability distributions, p(X,y) might change
- posterior probabilities p(y|X) might change.

Concept drift may be classified in terms of the [4] speed of change and the reason of change as shown in figure 1. When 'a set of examples has legitimate class labels at one time and has different legitimate labels at another time', it is real drift, i.e. reason of change[20], refers to changes in p(y|X).

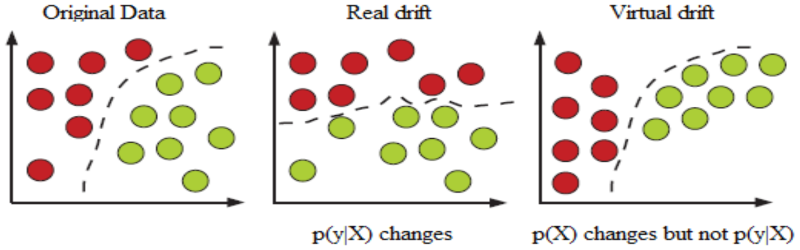

Fig 1: Types of drift: circles represent instances; different colors represent different classes[4]

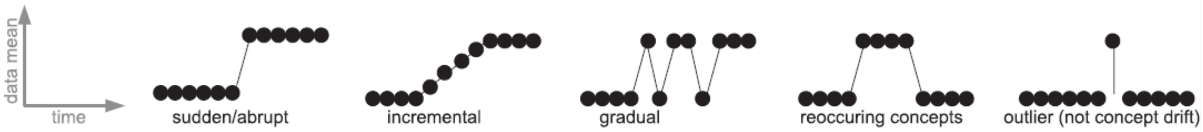

Fig 2: Patterns of concept change [4]

When 'the target concepts remain the same but the data distribution changes'[6], it is virtual drift, i.e. speed of change, refers to changes in p(X).

A drift can be sudden or abrupt, when concept switching is from one to another (refer figure 2)[4]. The concept change can be incremental, consisting of many intermediate concepts in between. Drift may be gradual; change is not abrupt, but goes back to previous pattern for some time. Concept drift handling algorithms should not mix the true drift with an outlier (blip) or noise, which refers to an anomaly. A recurring drifts is when new concepts that were not seen before, or previously seen concepts may reoccur after some time.

Detecting Concept changes:

The ways to monitor concept drift are as given below:

- Concept drift is monitored by checking with the data's probability distribution, since it changes with time.
- One can judge whether concept drift has happened, by monitoring and tracking the relevance between various sample characteristics or attributions.
- Concept drifts leads to changes in features of classification models.
- Classification accuracy can be taken into account while detecting concept drift on a given data stream. Recall, precision and F-measure are some of the accuracy indicators of classification

The arrival of the timestamp of single sample or block sample can be taken as an additional input attribute, to determine occurrence concept drift. It keeps a check on whether the classification rule has become outdated.

III. Concept Drift Detectors

This section discusses algorithms allowing to detect concept drift, known as concept drift detectors. They alarm the base learner, that the model should be rebuilt or updated.

DDM:

In the Drift Detection Method (DDM), proposed by Gama et al. uses Binomial Distribution[14]. For each point i in the sequence that is being sampled, the error rate is the probability of misclassifying (p_i), with standard deviation (s_i) given by eq 2-

$$s_i = \sqrt{\frac{pi(1-pi)}{i}} \qquad (2)$$

they store the values of p_i and s_i when $p_i + s_i$ reaches its minimum value during the process i.e. p_{min} and s_{min}. These values are used to calculate a warning level condition presented in eq. 3 and an alarm level condition presented in eq. 4 -

$$p_i + s_i \geq p_{min} + \alpha \, . \, s_{min} \quad \text{(warning level)} \qquad (3)$$

$$p_i + s_i \geq p_{min} + \beta . s_{min} \quad \text{(alarm level)} \qquad (4)$$

Beyond the warning the examples are stored in anticipation of a possible change of context. Beyond the alarm level, the concept drift is supposed to be true, the model induced by the learning method is reset, also p_{min} and s_{min}, and a new model is learnt using the examples stored since the warning level triggered. DDM works best on data streams with sudden drift as gradually changing concepts can pass without triggering the alarm level.

EDDM:

Baena-García et al. proposed a modification of DDM called EDDM [16]. The same warning- alarm mechanism, was used but instead of using the classifier's error rate, the distance-error-rate was proposed. They denote p'_i as the average distance between two consecutive errors and s'_i as its standard deviation. Using these values the new warning and alarm conditions are given by eq. 5 and eq. 6.

$$p'_i +2 . s'_i / p'_{max}+2.s'_{max} < \alpha \quad \text{(warning level)} \tag{5}$$

$$p'_i+3. s'_i/ p'_{max}+3.s'_{max} < \beta \quad \text{(alarm level)} \tag{6}$$

the values of p'_i and s'_i are stored when reaches its maximum value $p'_i +2 \ .s'_i$ (obtaining p'_{max} and s'_{max}). EDDM works better than DDM for slow gradual drift, but is more sensitive to noise. Another drawback is that it considers the thresholds and searches for concept drift when a minimum of 30 errors have occurred.

Adwin:

Bifet et al. proposed this method, that uses sliding windows of variable size, which are recomputed online according to the rate of change observed from the data in these windows[13]. The window(W) is dynamically enlarged when there is no clear change in the context, and shrinks it when a change is detected. Additionally, ADWIN provides rigorous guarantees of its performance, in the form of limits on the rates of false positives and false negatives. ADWIN works only for one-dimensional data. A separate window must be maintained for each dimension, for n-dimensional raw data, which results in handling more than one window.

Paired Learners:

The Paired Learners, proposed by Stephen Bach et al., uses two learners: stable and reactive[17]. The stable learner predicts based on all of its experience, while the reactive one predicts based on a window of recent examples. It uses the interplay between these two learners and their accuracy differences to cope with concept drift. The reactive learner can be implemented in two different ways; by rebuilding the learner with the last w(window size) examples, or by using a retractable learner that can unlearn examples.

Exponentially weighted moving average for Concept Drift Detection (ECDD):

Ross et al., proposed a drift detection method based on Exponentially Weighted Moving Average (EWMA)[15], used for identifying an increase in the mean of a sequence of random variables. In EWMA, the probability of incorrectly classifying an instance before the change point and the standard deviation of the stream are known. In ECDD, the values of success and failure probability(1 and 0) are computed online, based on the classification accuracy of the base learner in the actual instance, together with an estimator of the expected time between false positive detections.

Statistical Test of Equal Proportions (STEPD):

The STEPD proposed by Nishida et al., assumes that 'the accuracy of a classifier for recent W examples will be equal to the overall accuracy from the beginning of the learning if the target concept is stationary; and a significant decrease of recent accuracy suggests that the concept is changing'[18]. A chi-square test is performed by computing a statistic and its value is compared to the percentile of the standard normal distribution to obtain the observed significance level. If this value is less than a significance level, then the null-hypothesis is rejected, assuming that a concept drift has occurred. The warning and drift thresholds are also used, similar to the ones presented by DDM, EDDM, PHT, and ECDD.

DOF:

The method proposed by Sobhani et al. detects drifts by processing data chunk by chunk, the nearest neighbor in the previous batch is computed for each instance in the current batch and comparing their corresponding labels. A distance map is created, associating the index of the instance in the previous batch and the label computed by its nearest neighbor; degree of drift is computed based on the distance map. The average and standard deviation of all degrees of drift are computed and, if the current value is away from the average more than s standard deviations, a concept drift is raised, where s is a parameter of the algorithm. [10]This algorithm is more effective for problems with well separated and balanced classes.

IV. Statistical Tests For Concept Drift:

The design of a change detector is a compromise between detecting true changes and avoiding false alarms. This is accomplished by carrying out statistical tests that verifies if the running error or class distribution remain constant over time.

CUSUM test:

The cumulative sum algorithm[24], is a change detection algorithm that raises an alarm when the mean of the input data is significantly different from zero. The CUSUM input $\square_t$ can be any filter residual, for example, the prediction error from a Kalman filter. The CUSUM test is as follows-

$$g_{o\,=}\,0$$
$$g_t = \max\,(0, g_t\text{-}1 + \square_t - \upsilon)$$
$$\text{if } g_t > h \text{ then alarm and } g_t = 0 \qquad (7)$$

The CUSUM test is memoryless, and its accuracy depends on the choice of parameters υ and h.

Page Hinkley test: It is a sequential analysis technique, proposed by, that computes the observed values and their mean up to the current moment. The Page-Hinkley test[5] is given as -

$$g_{o\,=}\,0,\ g_t = g_t\text{-}1 + \square_t - \upsilon$$
$$G_t = \min(g_t)$$
$$\text{if } g_t - G_t > h \text{ then alarm and } g_t = 0 \qquad (8)$$

Geometric moving average test:

The Geometric Moving Average (GMA) test [25] is as below:

$$g_{o\,=}\,0$$
$$g_t = \lambda g_{t-1} + (1 - \lambda)\square_t$$
$$\text{if } g_t > h \text{ then alarm and } g_t = 0 \qquad (9)$$

The forgetting factor λ is used to give more or less weight to the last data arrived. The threshold h is used to tune the sensitivity and false alarm rate of the detector.

Statistical test:

CUSUM and GMA are methods those deal with numeric sequences. A statistical test is a procedure for deciding whether a hypothesis about a quantitative feature of a population is true or false. We test an hypothesis by drawing a random sample from the population in question and calculating an appropriate statistic on its items.

To detect change, we need to compare two sources of data, and decide if the hypothesis H_0 that they come from the same distribution is true. Otherwise, a hypothesis test will reject H_0 and a change is detected. The simplest way for hypothesis, is to study the difference from which a standard hypothesis test can be formulated.

$$\hat{\mu}_0 - \hat{\mu}_1 \in N(0, \sigma^2_0 + \sigma^2_1), \text{ under } H_0$$
$$\text{or, to make a } \chi^2 \text{ test,} \quad [(\hat{\mu}_0 - \hat{\mu}_1)^2 / \sigma^2_0 + \sigma^2_1] \in \chi^2(1), \text{ under } H_0$$

The Kolmogorov-Smirnov test (non-parametric) is another statistical test to compare two populations. The KS-test has the advantage of making no assumption about the distribution of data.

V. Concept Drift Handling

Kuncheva proposes to group ensemble strategies for changing environments[9] as follows:

- Dynamic combiners (horse racing): component classifiers are trained and their combination is changed using forgetting process.
- Updated training data : component classifiers in the ensemble are created incrementally by incoming examples.
- Updating the ensemble member : ensemble members are updated online or retrained with blocks of data.
- Structural changes of the ensemble : ensemble members are reevaluated and the worst classifiers are updated or replaced with a classifier trained on the most recent examples, with any concept change.
- Adding new features - The attributes used are changed, as an attribute becomes significant, without redesigning the ensemble structure.

The approaches to handle concept drifts includes single classifier and ensemble classifier approaches. The single classifiers are traditional learners that were modeled for stationary data mining and have the qualities of an online learner and a forgetting mechanism. Basically, ensemble classifiers are sets of single classifiers whose individual decisions are aggregated by a voting rule. The ensemble classifiers provide better classification accuracy as compared to the single classifiers due combined decision. They have a natural way of adapting to concept changes due to their modularity.

Streaming Ensemble Algorithm(SEA): The SEA[8], proposed by Street and Kim, changes its structure based on concept change. It is a heuristic replacement strategy of the weakest base classifier based on accuracy and diversity. The combined decision was based on simple majority voting and base classifiers unpruned. This algorithm works best for at most 25 components of the ensemble.

Accuracy Weighted Ensemble (AWE): In SEA, it is crucial to properly define the data chunk size as it determines the ensembles flexibility. The algorithm AWE, proposed by Wang et al., trains a new classifier C' on each incoming data chunk and use that chunk to evaluate all the existing ensemble members to select the best component classifiers. AWE is best suited for large data streams and works well for recurring and other drifts.

Adaptive Classifier Ensemble(ACE): To overcome AWE's slow drift reactions, Nishida proposed a hybrid approach in which a data chunk ensemble is aided by a drift detector, called Adaptive Classifier Ensemble (ACE), aims at reacting to sudden drifts by tracking the classifier's error rate with each incoming example, while slowly reconstructing a classifier ensemble with large chunks of examples.

Hoeffding option trees(HOT) and ASHT Bagging: Hoeffding Option Trees (HOT) provide a compact structure that works like a set of weighted classifiers, and are built in an incremental fashion. This algorithms[27] allows each training example to update a set of option nodes rather than just a single leaf. Adaptive-Size Hoeffding Tree Bagging (ASHT Bagging) diversifies ensemble members by using trees of different sizes and uses a forgetting mechanism. Compared to HOT, ASHT Bagging proves to be more accurate on most data sets. But both are time and memory expensive than option trees or single classifiers.

Accuracy Diversified Ensemble(ADE): The algorithm called Accuracy Diversified Ensemble (ADE)[22], not only selects but also updates components according to the current distribution. ADE differs from AWE in weight definition, the use of online base classifiers, bagging, and updating components with incoming examples. Compared to ASHT and HOT, we do not limit base classifier size, do not use any windows, and update members only if they are accurate enough according to the current distribution.

Accuracy Updated Ensemble(AUE): Compared to AWE, AUE1 [7] conditionally updates component classifiers. It maintains a weighted pool of component classifiers and predicts classes for incoming examples based on weighted voting rule. It substitutes the weakest performing ensemble member and new classifier is created with each data chunk of examples, also their weights are adjusted. It uses Hoeffding trees as component classifiers. Compared to AUE1, AUE2 introduces a new weighting function[22], does not require cross-validation of the candidate classifier, does not keep a classifier buffer, prunes its base learners, and always updates its components. It does not limit base classifier size and use any windows. The OAUE[23], tries to combine block-based ensembles and online processing.

VI. Datasets With Concept Drift

Artificial datasets give the ground truth of the data, however, real datasets are more interesting as they correspond to real-world applications where the algorithms' usability is tested[22].

6.1 Real datasets:

- Forest Covertype, obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS) data, contains 581, 012 instances and 54 attributes.
- Poker-Hand consists of 1,000, 000 instances and 11 attributes.
- Electricity dataset, collected from the Australian New South Wales Electricity Market, contains 45, 312 instances.
- Airlines Dataset contains 539,383 examples described by seven attributes.
- Ozone level detection data set consists of 2,534 entries and is highly unbalanced (2% or 5% positives depending on the criteria of "ozone days").

6.2 Synthetic datasets:

The synthetic datasets allow us to analyze how the methods deal with the types of drift included in the datasets, as it is known in advance when the drifts begin and end. For abrupt or sudden drifts, Stagger, Gauss, Mixed2 can be used. The Waveforrm, LED generator or Circles dataset best suited for gradual drifts. Hyperplane dataset works well for both gradual and incremental drift. Radial basis function(RBF) can also be used for incremental drift, and blips can also be incorporated.

Applications

This sections describes various real-life problem [11,12] in different domains related to the concept drifts in the data generated from these real domains.

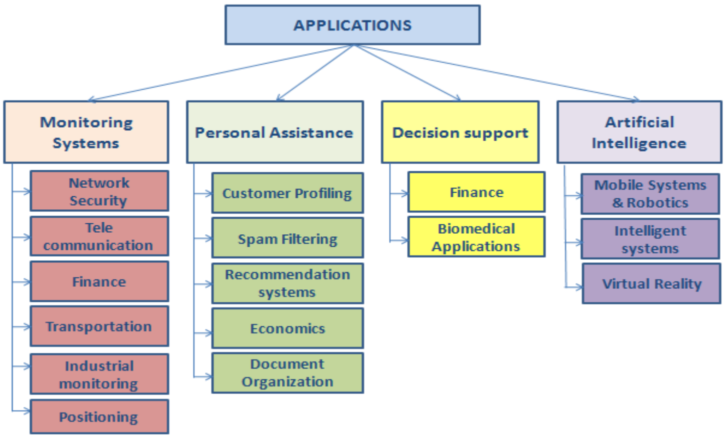

Fig 3: Applications of Real-domain concept drift

Monitoring and control often employs unsupervised learning, which detects abnormal behavior. In monitoring and control applications the data volumes are large and it needs to be processed in real time.

Personal assistance and information applications mainly organize and/or personalize the flow of information. the class labels are mostly ‘soft’ and the costs of mistake are relatively low.

Decision support includes diagnostics, evaluation of creditworthiness. Decision support and diagnostics applications usually involve limited amount of data. Decisions are not required to be made in real time but high accuracy is essential in these applications and the costs of mistakes are large.

Artificial intelligence applications include a wide spectrum of moving and stationary systems, which interact with changing environment. The objects learn how to interact with the environment and since the environment is changing, the learners need to be adaptive.

VII. Conclusion

This paper describes about the problem of concept drift. It summarizes the need, types and reasons for concept change. The various concept drift detection methods viz. DDM, EDDM, Paired learners, ECDD, ADWIN, STEPD and DOF are discussed and methods it adopts to detect concept change. To identify if concept drift has occurred, statistical tests like CUSUM, Page-Hinkley and GMA test are explained. Various classifier approaches, especially, ensemble classifiers provide better accuracy in case of concept change. The ensemble classifiers SEA, AWE, ACE, ADE, HOT, ASHT, AUE adapt according to the drift that occurs, yielding good classifier accuracy. Later, applications and the datasets, real and synthetic, suited for various concept drifts can be used to check the adaptability of any algorithm handling concept drift.

In future, we can enhance the classification performance of the ensemble algorithms discussed above, by adapting it to various drifts and diversity.

References

[1]. P. M. Goncalves, Silas G.T. de Carvalho Santos, Roberto S.M. Barros, Davi C.L. Vieira, (2014) "Review: A comparative stu dy on concept drift detectors", A International Journal: Expert Systems with Applications,8144–8156.

[2]. L. L. Minku and Xin Yao(2011), "DDD: A New Ensemble Approach For Dealing With Concept Drift", IEEE TKDE, Vol. 24, pp. 619 - 633.

[3]. M. G. Kelly, D. J. Hand, and N. M. Adams(1999), "The Impact of Changing Populations on Classifier Performance", In Proc. of the 5th ACM SIGKDD Int. Conf. on Knowl. Disc. and Dat. Mining (KDD). ACM, 367–371.

[4]. J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia(2014), "A Survey on Concept Drift Adaptation ", ACM Computing Surveys, Vol. 46, No. 4, Article 44.

[5]. Mouss, H., Mouss, D., Mouss, N., Sefouhi, L.(2004), "Test of Page-Hinkley, an Approach for Fault Detection in an Agro-Alimentary Production System", 5th Asian Control Conference, IEEE Computer Society, vol. 2, pp. 815--818.

[6]. S. Delany, P. Cunningham, A. Tsymbal, and L. Coyle. (2005),"A Case-based Technique for Tracking Concept Drift in Spam filtering", Knowledge-Based Sys. 18, 4–5 , 187–195.

[7]. D. Brzezinski and J. Stefanowski(2011), “Accuracy updated ensemble for data streams with concept drift,” Proc. 6th HAIS Int. Conf. Hybrid Artificial Inteligent. Syst., II, pp. 155–163.

[8]. W. N. Street and Y. Kim(2001), “A streaming ensemble algorithm (SEA) for large-scale classification,” in Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 377–382.

[9]. Ludmila I. Kuncheva(2004), "Classifier ensembles for changing environments", Multiple Classifier Systems, Lecture Notes in Computer Science, Springer , vol. 3077, pages 1–15.

[10]. Sobhani P. and Beigy H.(2011), "New drift detection method for data streams", Adaptive and intelligent systems, Lecture notes in computer science, Vol. 6943, pp. 88–97.

[11]. D Brzezinski, J Stefanowsk(2011), "Mining data streams with concept drift " Poznan University of Technology Faculty of Computing Science and Management Institute of Computing Science.

[12]. I Žliobaite (2010), "Adaptive Training Set Formation", Doctoral dissertation Physical sciences, informatics (09P) Vilnius University.

[13]. A Bifet(2009), "Adaptive Learning and Mining for Data Streams and Frequent Patterns", Doctoral Thesis.

[14]. J Gama, P Medas, G Castillo and Pedro Rodrigues(2004), "Learning with Drift Detection", Lecture Notes in Computer Science, Vol. 3171, pp 286-295.

[15]. G. J. Ross, N. M. Adams, D. Tasoulis, D. Hand(2012), "Exponentially weighted moving average charts for detecting concept drift", International Journal Pattern Recognition Letters, 191-198.

[16]. M Baena-Garcia, J Campo-Avila, R Fidalgo, A Bifet, R Gavaldµa and R Morales-Bueno(2006), "Early Drift Detection Method", IWKDDS, pp. 77–86.

[17]. S. H. Bach and M. A. Maloof (2008), "Paired Learners for Concept Drift", Eighth IEEE International Conference on Data Mining, pp. 23-32.

[18]. K. Nishida(2008), "Learning and Detecting Concept Drift", A Dissertation: Doctor of Philosophy in Information Science and Technology, Graduate School of Information Science and Technology, Hokkaido University.

[19]. D Brzezinski, J Stefanowski(2012), "From Block-based Ensembles to Online Learners In Changing Data Streams: If- and How-To", ECML PKDD Workshop on Instant Interactive Data Mining, pp. 60–965.

[20]. J. Kolter and M. A. Maloof (2007), "Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts", Journal of Machine Learning Research 8, 2755-2790.

[21]. P. B. Dongre, L. G. Malik(2014), " A Review on Real Time Data Stream Classification and Adapting To Various Concept Drift Scenarios", IEEE International Advance Computing Conference (IACC), pp. 533-537.

[22]. D Brzezinski, J Stefanowski (2014),"Reacting to Different Types of Concept Drift:The Accuracy Updated Ensemble Algorithm" IEEE Transactions On Neural Networks And Learning Systems, Vol. 25, pp. 81-94.

[23]. D Brzezinski, J Stefanowski(2014), "Combining block-based and online methods in learning ensembles from concept drifting data streams", An International Journal: Information Sciences 265, 50–67.

[24]. E. S. Page.(1954) Continuous inspection schemes. Biometrika, 41(1/2):100–115.

[25]. S. W. Roberts(2000), "Control chart tests based on geometric moving averages", Technometrics, 42(1):97–101.

[26]. R. Elwell and R. Polikar(2011), “Incremental learning of concept drift in nonstationary environments,” IEEE Trans. Neural Netw., vol. 22, no. 10, pp. 1517–1531.

[27]. A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà(2009),“New ensemble methods for evolving data streams,” in Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 139-148.