Handling Concept Drift in Data Streams by Using Drift Detection Methods

Malini M. Patil

Abstract The growth and development of the information and communication technology of the present era resulted in huge amount data generation. It is found that the rate of data distribution is very high. The data which is generated with varying distributions is referred to as data stream. Few examples to quote, data generated with regard to applications related to mobile networks, sensor networks, network traffic monitoring and network traffic management, etc. It is found that, the data generation process often change with respect to data distribution for any kind of concept, i.e. application which is referred to as concept drift. Handling concept drift is a challenging task. It is impossible to develop a model as it will be inconsistent in nature because of continuous change. The present work emphasises on handling the concept drifts, using different drift detection methods using Massive Online Analysis Framework. The important feature of the present study is varying size of a data stream (50,000–250,000). Totally the Concept Drift is handled using 11 drift detection methods using 2 stream generators abrupt and gradual under this frame work respectively.

Keywords Concept drift · Data streams · Data distribution · Drift detection
Massive online analysis framework

1 Introduction

In the recent past the data generation process is found to be very huge, with varying data distributions. Few real-time examples which are listed under this feature are weather forecasting data applications related to mobile networks, sensor networks, network traffic monitoring, and network traffic management, social network data,

M. M. Patil (✉)
Department of Information Science and Engineering,
J S S Academy of Technical Education, Bengaluru, India
e-mail: drmalinimpatil@gmail.com

V. E. Balas et al. (eds.), *Data Management, Analytics and Innovation*,
Advances in Intelligent Systems and Computing 839,
https://doi.org/10.1007/978-981-13-1274-8_12

etc. Data which is generated with varying distributions is called as a data stream or a streaming data. Data Stream is far different from traditional databases.

Some of the features of data streams are identified as: (i) Data streams are huge in size. (ii) Data streams are ubiquitous in nature. (iii) Data streams require fast response. (iv) It is not possible to access the data streams randomly. (v) They require limited memory for storage. (vi) Require highly sophisticated techniques for mining. The challenges of data streams are: (i) Data stream processing algorithms does not permit multiple scans when compared to traditional data mining algorithms. (ii) Faster mining methods are to be employed with respect to speed of incoming data for faster response. (iii) Handling the change in data distribution is the major challenge for mining data streams. From the literature it is identified that the data streams can be studied under two headings (1) *Static streams* (2) *Evolving streams*. History data or regular bulk arrivals are termed as Static streams e.g. queries on data warehouses. Real time data which gets updated constantly are termed as evolving data streams [1] e.g. stock market data, and sensor data.

1.1 Concept Drift

Weather forecasting data [2, 3] which is quoted as one of the important real-time example to understand the feature of the data stream is mainly used to explain the concept drift. As it is known that the forecast variation is seasonal and the data distribution changes are likely to occur continuously. Another example is customer buying pattern in an inventory system. It is clearly observed that the customer buying preferences change quite often with respect to time. From both the examples it is clear that the cause of the change in weather forecast and change in customer preferences is hidden. Changes are hidden and unpredictable. The study reveals that an effective learner should be designed so that the learner can detect and able to track the hidden and unpredictable changes. One of the important problems identified in handling the concept drift is differentiation between noise and concept drift. Normally it is advised to design the ideal concept drift handling system which can have a method to adapt to concept drift by differentiating with noise.

a. **Types of concept drifts**

   The different types of concept drifts identified in the literature [4] are listed as follows:

   a. **Abrupt concept drift**: This type of drift refers to sudden or instant changes that occur. e.g., Changes on seasonal demand on sales.
   b. **Incremental concept drift**: This type of drift occurs when the values of variables are changed slowly over a period of time. e.g. slow increase in prices.
   c. **Gradual concept drift**: This type of drift occurs when variables change their class distribution slowly over time. e.g. spam data.

d. **Recurring concept drift**: This type of drift represents temporary changes that occur in data streams. They can be reverted back to their original state after some time. e.g. Trends in market.

b. **Drift detection**
Drift detection is a technique used to determine concept drift between two or more time periods. *Drift detector* is an algorithm that accepts input as stream of instances of varying sizes. The output of an algorithm is identification of concept drift, i.e. the detection of change in the distribution of the data. Drift detection or Change detection is a challenging task which consists of detecting the true changes. Study reveals that different drift-detection techniques are available. But the present work focuses mainly Handling Concept Drift in data streams by using Different Drift Detection Methods in Massive Online Analysis Framework.

2 Literature Survey

From the study of literature survey it is found that STAGGER [4] is one of the important bench mark data set which was able to handle concept drift. Later it was found to be the most popular bench mark data set. The study also reveals that many learning algorithms were used in handling concept drifts. They are rule-based learning, decision trees, Naïve Bayes, Radial Basis Functions instance-based learning. The interesting point to be high lighted here is that the lazy learning techniques are also found to be appropriate to handle concept drift which is presented very efficiently in [5]. The authors have proposed a case-based system for spam filtering using dynamic approach. In [6] the authors present the new ensemble learning methods. In [7] the authors elaborated on the study of concept drift in continuos domains. In [8] the authors present a very novel approach based on the learning method which is incremental. The method is based on a distributed concept description which is composed of a set of weighted, symbolic characterizations. The method utilizes previously acquired concept definitions in subsequent learning by adding an attribute for each learned concept to instance descriptions. Another important aspect of detecting concept drift is proposed in [9] in which the authors have developed a classification model of data stream. The adaptive window based approach proposed by authors of the paper [10] can detect different types of drift, is based on processing data chunk by chunk and measuring differences between two consecutive batches, as drift indicator. The experimental results show that the proposed method is capable to detect drifts and can approximately find concept drift locations. On the other side the platform used for running experiments was a break through contribution to the research developed by [11–16]. It is Massive Online Analysis framework. The authors have used the frame work extensively for the mining of data streams using MOA frame work. Perceptron learning model on evolving streams [17], study of recommender systems [18], performance analysis of hoeffding trees [19], frequent itemset mining on data streams [20], regression modelling using IBL streams [21] and mining data streams with concept drift [22] are some of the important works of the authors.

3 Drift Detection Methods

The MOA release 2014.04 provides 11 different drift detection methods. This section provides a brief note on all the methods.

a. *Adwin Change detector (DD1)*: ADWIN [10, 16] stands for Adaptive Windowing and uses sliding windows. Sliding window is one of the key data processing models in data streams. The size of the sliding window is application or machine dependent. Adwin automatically readjusts the window size with respect to change in data distribution and the window size is recomputed online.
b. *CusumDM (DD2)*: CusumDM stands for Cumulative Sum detection method [23–25] CUSUM is published in Biometrika in the year 1954, E. S. Page. It is a sequential analysis technique. In drift detection it is used for monitoring change detection.
c. *DDM (DD3)*: DDM stands for Drift Detection Method. It uses a Binomial Distribution [26] to detect the changes. Binomial distribution gives the general form of the probability for the random variable that represents the number of errors in a sample of 'n' examples. DDM handles the classification errors produced by the learning model during prediction.
d. *EDDM (DD4)*: EDDM stands for Early Drift Detection Method (EDDM) [27]. This method is developed as an improvement over DDM to detect the concept drift. The basic idea is to find the distance between classification errors to detect change. It can detect the change without increasing the rate of false positives. It is also capable of detecting the slow gradual changes. The study reveals that it will improve the predictions.
e. *EWMAChartDM (DD5)*: EWMACHARTDM [28] stands for Exponentially Weighted Moving Average Chart Detection Method. It is a new modular approach for detecting concept drift. The method is designed in such a way that it is able to monitor the misclassification rate of a streaming classifier. It can also detect the change without increasing the rate of false positives during prediction.
f. *GMA DM (DD6)*: GMADM stands for Geometrical Moving Average Detection Method [29]. It is based on the concept of assigning weights to the observations for detecting changes in data streams. The method of assigning weights is based on geometric progression. The latest first observation is assigned with the greatest weight. The previous weights assigned to the observations were found to decrease in geometric progression.
g. *HDDM-A-Test (DD7) and HDDM-W-Test (DD8)*: HDDM-A-Test stands for The Hoeffding-based Drift Detection Method. It involves moving averages to detect abrupt changes (HDDM) [30]. HDDM-W-Test stands for The Hoeffding-based Drift Detection Method which is also a window based approach as mentioned earlier. It mainly detects gradual changes using weighted moving averages.

h. *PageHinkley DM (DD9)*: PageHinkley test [25, 31] is the sequential analysis technique typically used for monitoring concept drift detection. It allows efficient detection of changes in the normal behaviour of a process which is established by a model.
i. *SeqChange1DriftDetector (DD10) and SeqChange2DriftDetector (DD11)*: Sequential change-point detection [16, 31] is concerned with the design and analysis of techniques for online detection of a concept drift to a tolerable limit on the risk of a false detection. It is also referred to as quickest change detection method.

4 Methods and Models

This section presents the methodology used in this paper to compare the above mentioned drift detectors which are briefly explained in the previous section. The section covers the brief introduction of Massive Online Analysis framework (MOA), the process of evaluation in MOA, performance evaluators of MOA, Data sources in MOA.

4.1 Massive Online Analysis (MOA) Framework

Massive online analysis frame work [15] is an open source software environment to handle massive evolving data which is potentially infinite. It is also supported with features used for implementing algorithms and running experiments for evolving data streams. The present work uses MOA-14 for running the algorithms and conducting the experiments. MOA-14 is designed in such a way that it can meet the challenges of handling both the online and offline data streams. It can also handle real world data sets. It consists of offline and online algorithms for classification and clustering. It also consists of tools for evaluation. It is able to meet the important challenges of data streams. Study reveals that many of the works found in the literature establish the challenges of the data streams. MOA mainly permits the evaluation of data stream learning algorithms on massive data streams under explicit memory limits.

4.2 Methodology

The sequence of steps involved in configuring the MOA framework for detecting the concept drift using 11 Drift Detection Methods is as shown in Fig. 1.

The main steps involved in the drift detection method are listed as follows: (a) Selection of the generator. (b) Selection of the performance evaluator.

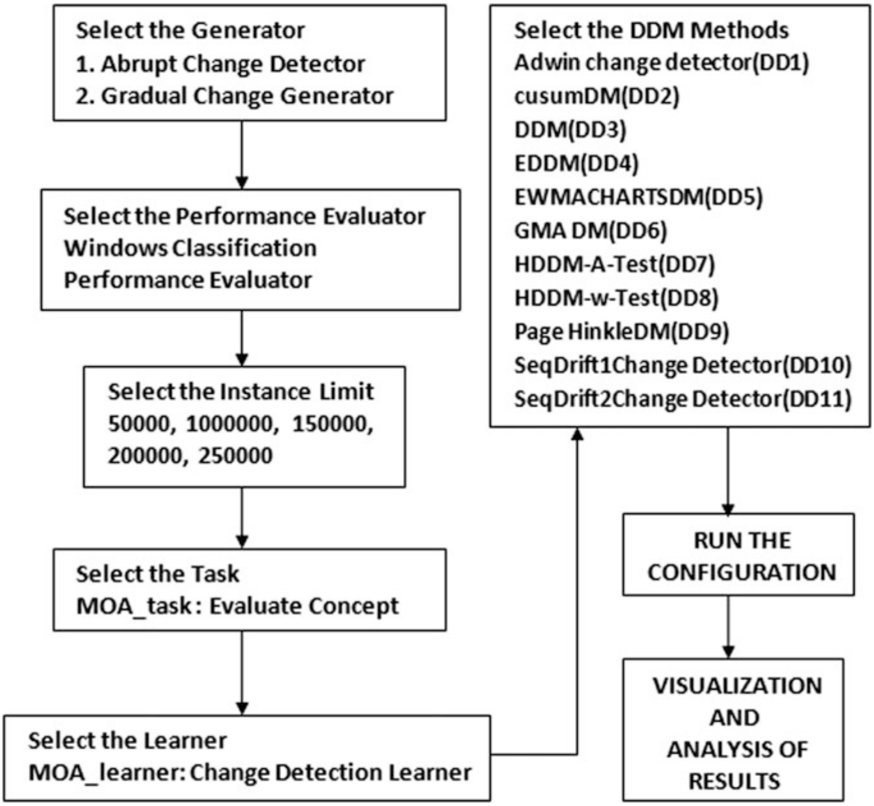

Fig. 1 Methodology used in drift detection method

(c) Selection of the instance limit. (d) Selection of the task. (e) Selection of the learner. (f) Selection of the drift detection method. (g) Visualization of the results. The other details of the steps are shown in Fig. 1. MOA is embedded with two types of concept drifts namely *abrupt* and *gradual.* The present work uses both the types. The performance evaluators available in MOA are *windows classification performance evaluator* and *basic classification performance evaluator*. The present work uses the windows method. Selection of instance is an auto generation method embedded in MOA. For the evaluation purpose the instance limit selected is 50,000–250,000. In the moa_task selection process selects *evaluate_concept drift* as the task which is the main purpose of the present work. Naïve Bayes is the learner used for handling the concept drift. The Whole selection process mentioned above is kept same for both the types of concept drifts and only the drift detection method is varied every time before running the experiment. The time taken to run the configuration is tabulated accordingly.

5 Experiments and Results

The configuration of the drift detection method using MOA framework is as shown in Fig. 2.

Experiments are carried out on the 11 drift detection methods mentioned in Sect. 3 by using abrupt and gradual drift generators. Naïve Bayes learner is used for the evaluation purpose. The results are tabulated in Tables 1, and 2 respectively. The graphical representation of the results is shown in Figs. 3, and 4 respectively.

The glance at Table 1 reveals that the behaviour of all the drift detectors is very interesting in case of gradual generator method. Following are the observations made in the analysis.

- For an instance size of 5000, CUSUMDM (DD2) performs well. DD2 takes minimum execution time (1.93 s) where as SEQUENTIAL CHANGE 1 DETECTOR (DD10) takes maximum execution time (5.04 s).
- For an instance size of 100,000, 150,000, 200,000 SEQUENTIAL CHANGE 2 DETECTOR (DD11) takes maximum execution time (14.76, 50.48, 45.29 s) and EDDM (DD4) takes minimum execution time (4.12, 5.32, 8.05 s).
- For an instance size of 250,000, SEQUENTIAL CHANGE 2 DETECTOR (DD11) takes minimum execution time (9.3 s) SEQUENTIAL CHANGE 1 DETECTOR (DD11) takes maximum execution time (21.96 s).
- EDDM (DD4) performs better with respect to all the drift detectors for varying instance sizes.
- The graphical representation of behaviour of all the drift detectors in case of gradual generator method is as shown in the Fig. 3 and is self-explanatory.

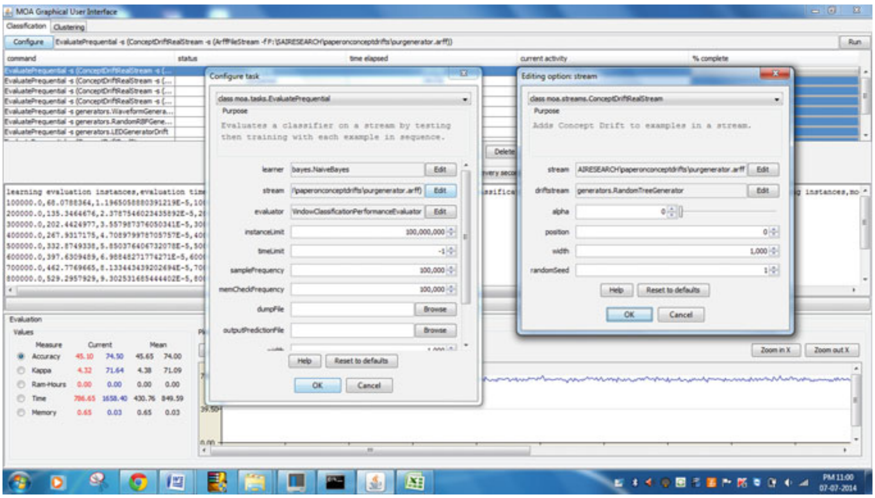

Fig. 2 Configuration window in MOA

Table 1 Time (in sec) recorded for gradual generator method

Drift detector	Instance size				
	50,000	100,000	150,000	200,000	250,000
ADWIN Change detector (DD1)	2.39	5.66	9.09	9.59	14.76
CUSUMDM (DD2)	1.93	5.13	6.58	10.90	11.9
DDM (DD3)	3.00	6.43	5.35	8.30	11.23
EDDM (DD4)	2.37	4.12	5.32	8.05	10.83
EWMAChartDM (DD5)	2.71	4.93	6.65	9.63	12.18
GMA DM (DD6)	2.73	4.96	6.55	9.70	12.60
HDDM-A-Test (DD7)	2.78	5.05	10.92	10.48	11.89
HDDM-W-Test (DD8)	2.96	5.94	12.95	11.01	12.96
PAGEHINKLEY DM (DD9)	2.89	5.34	10.87	10.59	13.37
SEQCHANGE1DRIFTDETECTOR (DD10)	5.04	9.92	21.61	16.61	21.96
SEQCHANGE2DRIFTDETECTOR (DD11)	4.37	14.76	50.48	45.29	9.30

Table 2 Time (in sec) recorded for abrupt generator method

Drift detector	Instance size				
	50,000	100,000	150,000	200,000	250,000
ADWIN Change detector (DD1)	2.48	5.23	6.86	10.92	11.31
CUSUMDM (DD2)	2.32	5.16	8.03	10.22	13.79
DDM (DD3)	1.92	4.93	6.46	8.6	13.54
EDDM (DD4)	1.67	4.21	5.05	8	8.42
EWMAChartDM (DD5)	2.04	4.96	5.44	9.94	13.48
GMA DM (DD6)	2.40	4.96	6.65	9.63	12.68
HDDM-A-Test (DD7)	0.05	5.18	6.85	10.14	15.33
HDDM-W-Test (DD8)	0.05	5.46	8.13	9.98	13.10
PAGEHINKLEY DM (DD9)	0.05	5.41	6.74	10.31	10.89
SEQCHANGE1DRIFTDETECTOR (DD10)	0.05	11.17	21.62	28.59	14.38
SEQCHANGE2DRIFTDETECTOR (DD11)	0.05	5.37	7.00	10.42	9.97

The glance at Table 2 reveals that the behaviour of all the drift detectors is also very interesting in case of abrupt generator method. Following are the observations made in the analysis.

- For an instance size of 5000, the drift detectors DD7 to DD11 performs well and are consistent with the minimum execution time (0.05 s). Adwin Change detector (DD1) takes maximum execution time (2.48 s).

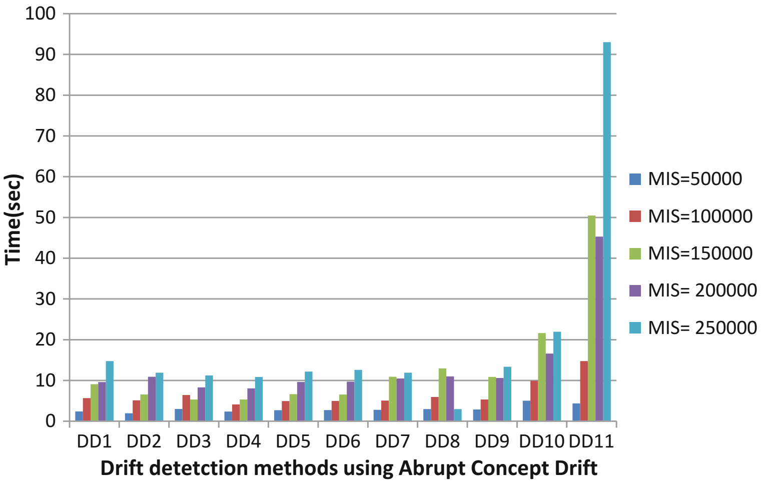

Fig. 3 Graph of drift detection methods using abrupt concept drift for varying instance sizes

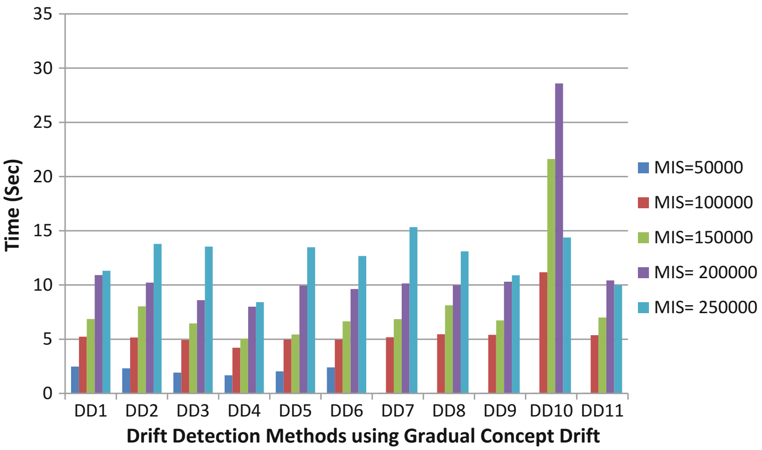

Fig. 4 Graph of drift detection methods using gradual concept drift for varying instance sizes

- For instance sizes 100,000, 150,000, 200,000, 250,000 it is very interesting to note that EDDM (DD4) performs very well with minimum execution time (4.21, 5.05, 8 and 8.42 s) respectively.
- For an instance sizes of 100,000, 150,000 and 200,000 SEQUENTIAL CHANGE 1 DETECTOR (DD10) takes maximum time (11.17, 21.62 and 28.59 s) respectively.
- For an instance size of 250,000 HDDM (DD7) take maximum execution time (15.33 s) and EDDM (DD4) takes minimum time (8.42 s).

- EDDM (DD4) performs better with respect to all the drift detectors for varying instance sizes.
- The graphical representation of behaviour of all the drift detectors in case of abrupt generator method is as shown in the Fig. 4 and is self-explanatory.

6 Conclusions

The present work is mainly emphasis on the understanding of the concept drift in data streams by using different drift detection methods for gradual and abrupt generators in MOA framework which is concluded with the following observations.

- The present analysis uses drift detectors namely Adwin Change Detector (DD1), CUSUMDM (DD2), Drift Detection Method (DD3), EDDM (DD4), EWMAChartDM (DD5), Geometric Moving Average DM (DD6), HDDM-A-Test (DD7) and HDDM-W-Test (DD8), Page HinkleyDM (DD9), Sequential Change 1 Detector (DD10), Sequential Change 2 Detector (DD11).
- The maximum instance size used in the analysis varies from 500,000, 100,000, 150,000, 200,000, and 250,000.
- Naïve Bayes learner is used for the evaluation purpose.
- The performance of EDDM (DD4) is found to be good with respect to abrupt and gradual generators especially in case of instance sizes 100,000, 150,000 and 200,000.
- Another important observation is for an instance size of 500000 the drift detectors HDDM-A-Test (DD7) and HDDM-W-Test (DD8), Page HinkleyDM (DD9), Sequential Change 1 Detector (DD10), Sequential Change 2 Detector (DD11) perform well and are consistent with the minimum execution time (0.05 s).
- The results of the present investigation are unique in handling concept drift in MOA frame work and also they provide an excellent platform for future investigation.

References

1. Aggarwal, C. (2007). *Data streams: Models and algorithms.* Series: Advances in database systems (Vol. 31, XVIII, p. 354). Berlin, Heidelberg: Springer (ebook).
2. Gaber, M. M., Zaslavsky, A., & Krishnamurthy, S.(2005). *Data streams: Models and methods* (Vol. 31, pp. 39–59). Berlin, Heidelberg: Springer.
3. Gao, J., Fan, W., Han, J., & Yu, P. S. (2007). General framework for mining concept-drifting data streams with skewed distributions. In *SIAM International Conference on Data Mining* (pp. 3–14), Minneapolis.
4. Brzesinski, D. (2010). *Mining data streams with concept drift (Ph.D. thesis).*

5. Cunningham, P., Nowlan, N., Delany, S. J., & Haahr, M. (2003). A case-based approach to spam filtering that can track concept drift. In *The Proceedings of ICCBR-2003 Workshop on Long-Lived CBR System*.
6. Kolter, J. Z., Maloof, M. A. (2003). Dynamic weighted majority: A new ensemble method for tracking concept drift. In *3rd IEEE International Conference on Data Mining ICDM-2003* (pp. 123–130). IEEE CS Press.
7. Kubat, M., & Widmer, G. (1994). *Adapting to drift in continuous domains* (Technical Report). Vienna: Austrian Research Institute for Artificial Intelligence.
8. Schlimmer, J. C., & Granger, R. H. (1986). Incremental learning from noisy data. *Journal of Machine Learning, 1*(3), 317–354.
9. Xiaofeng, L., & Weiwei, G. (2014). SERSC study on a classification model of data stream based on concept drift. *International Journal of Multimedia and Ubiquitous Engineering, 9* (5), 363–372. http://dx.doi.org/10.14257/ijmue.2014.9.5.37. ISSN: 1975-0080.
10. Bifet, A., & Gavalda, R. *Learning from time changing data with adaptive windowing*. Univeritat Poliyecnica De Catalunya.
11. Bifet, A., Holmes, G., Kirby, R., & Pfahringer, B. (2010). MOA: Massive online analysis. *Journal Machine Learning Research*, 1601–1604.
12. Bifet, A., Holmes, G., Kirby, R., & Pfahringer, B. (2011). MOA: Massive online analysis. *Journal of Machine Learning Research*, 1601–1604.
13. Bifet, A., Kirkby, R., Kranen, P., & Reutemann, P. (2009). *Massive online analysis, technical manual*. Hamilton, New Zealand: University of Waikato.
14. Bifet, A., & Kirkby, R. (2009). *Data stream mining: A practical approach* (Technical report). New Zealand: The University of Waikato.
15. https://moa.cms.waikato.ac.nz/
16. Bifet, A., & Gavaldà, R. (2009). Adaptive learning from evolving data streams. *Advances in Intelligent Data Analysis,* 8, 249–260. Berlin, Heidelberg: Springer.
17. Srimani, P. K., & Patil, M. M. (2012). Simple perceptron model (SPM) on evolving streams in MDM. *International Journal of Neural Networks, 2*(1), 20–24. E-ISSN 2249-2771.
18. Patil, M. M. (2015). A comprehensive study of recommender systems. *International Journal of Advancements in Engineering Research, 10*(86), 332–337.
19. Srimani, P. K., & Patil, M. M. (2015). Performance analysis of hoeffding trees in MDM using MOA framework. *International Journal of Data Mining, Modeling and Management, 7*(4), 293–313. http://dx.doi.org/10.1504/IJDMMM.2015.073865.
20. Srimani, P. K., & Patil, M. M. (2015). Frequent item set mining using INC_MINE in massive on line analysis framework. *Science Direct Journal of Elsevier Publication, 45*, 133–142.
21. Srimani, P. K., & Patil, M. M. (2014). Regression modeling using IBLSTREAMS. *Indian Journal of Science and Technology, 7*(6), 864–870. Print ISSN 0974-6846, Online ISSN 0974-5645.
22. Srimani, P. K., & Patil, M. M. (2016). Mining data streams with concept drift in massive online analysis framework. *WSEAS transactions on computers, 15*(#14), 133–139. Article is in the press. ISSN/E-ISSN 1109-2750/2224-2872.
23. Barnard, G. A. (1959). Control charts and stochastic processes. Journal of the Royal Statistical Society. *B (Methodological), 21*(2), 239–271. JSTOR 2983801.
24. Grigg, Farewell, V. T., Spiegelhalter, D.J., et al. (2003). The use of risk-adjusted CUSUM and RSPRT charts for monitoring in medical contexts. *Statistical Methods in Medical Research, 12*(2): 147–170. https://doi.org/10.1177/096228020301200205.pmid12665208.
25. Page, E. S. (1954). Continuous inspection schemes. *Biometrika, 41*, 100–115.
26. Gama, J., & Bifet, A. (2014). *A survey on concept drift adaptation*. Portugal: University of Porto.
27. Gavaldμa, R., & Morales-Bueno, R.(2011). *Early drift detection method manuel.* Spain: University of Malaga.

28. Rossa, G. J., Adamsa, N. M., Tasoulisa, D. K., & Handaa, D. J. (2012). *Exponentially weighted moving average charts for detecting concept drift*. Department of Mathematics, Imperial College, London SW7 2AZ, UK.
29. Roberts, S. W. (2012). In M. Brama & M. Petridis (Eds.), Control chart tests based on geometric moving averages. *Research Development in intelligent systems, XXIX*, 97–101. https://doi.org/10.1007/978-1-4471-4739-8-6 .
30. Frías-Blanco, I., Campo-Ávila, J. Ramos, G., Bueno, R., 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.
31. Moustakides, G. V. (2008). Sequential change detection revisited. *The Annals of Statistics 2008, 36*(2), 787–807. https://doi.org/10.1214/009053607000000938.