A Comparative Analysis of Concept Drift Detection Methods with a Systematic and Innovative Approach of Method Selection

M.SC. FABIAN GERZ, DR.-ENG. LOUI AL-SHROUF
and PROF. DR.-ENG. MOHIEDDINE JELALI

ABSTRACT

One of the most significant challenges in data-driven modeling of complex systems is dealing with concept drift, i.e., the unpredictable changes in the underlying data distribution over time. In this work, nine concept drift detection (CDD) methods are evaluated with respect to different types of concept drift, including abrupt, gradual, incremental, and real concept drift, for both supervised and unsupervised application scenarios. For the supervised case, the methods EDDM, $FHDDMS_{add}$, MDDM-E, and EFDT are compared against a classification without change detection using Naïve Bayes as the base classifier. In the unsupervised application scenario, CluStream, ClusTree, DenStream, StreamKM++, and D-Stream are evaluated. The experiments are conducted using the Massive Online Analysis (MOA) evaluation platform, and the performance of each method is measured in terms of classification accuracy, memory consumption, and computation time. This empirical research shows that classification accuracy can be improved by 20% by implementing a CDD method, highlighting the importance of CDD in SHM data streams. However, there is no single method that proves to be superior in all scenarios, and the choice depends on the characteristics of the considered data stream and application requirements. Selecting the appropriate CDD method from the approximately 340 different methods found in the literature is not a trivial task and can lead to suboptimal selection. To tackle this issue, an innovative approach is proposed to assist researchers and practitioners find the appropriate CCD method for their application.

INTRODUCTION

Information extraction and classification are two examples of the diverse Machine Learning (ML) applications. In the context of Structural Health Monitoring (SHM), this means an extension of the capabilities in all stages of damage identification [1]. However, many of the implemented ML approaches neglect environmental and operational factors (EOFs) such as temperature effects or traffic volume, but rely only on severe damage to the structures [1, 2]. Traditional algorithms and models assume

M.Sc.-Eng. Fabian Gerz, Dr.-Eng. Loui Al-Shrouf, Prof. Dr.-Eng. Mohieddine Jelali, Cologne Lab for Artificial Intelligence and Smart Automation – CAISA, TH Köln – University of Applied Sciences, Betzdorfer street 2, 50679 Köln (Cologne), Germany.

that the environment is static and, consequently, the statistical distribution of the data does not change over time. However, this assumption does not apply to real-world scenarios and, thus, their data. These so called static methods have the disadvantage that the accuracy of their prediction quality decreases with the occurrence of a changed data source (concept drift). Therefore, an adaptation of the methods for ML models is required to account for this change. Formally, concept drift can be defined as follows: Within a time interval $[0, t]$ data instances $d_i = (X_i, Y_i)$ form a dataset $s_{0,t} = [d_0, \ldots, d_t]$, where X_i is a feature vector and Y_i is the corresponding class label. Moreover, the dataset $s_{0,t}$ follows a distribution $F_{0,t}$. If this changes with respect to the distribution of the next time step, concept drift is present [3–6].

$$F_{0,t}(X|y) \neq F_{t+1,\infty}(X|y), \text{ denoted as } \exists t: P_t(X|y) \neq P_{t+1}(X|y) \quad (1)$$

In 1986, the term concept drift was first shaped by Schlimmer and Granger [5]. They developed an adaptive learning algorithm using an incremental approach [7]. Based on this, numerous methods and algorithms have been developed to address the concept drift problem [8–10]. In addition, a number of comprehensive studies exist [3, 4, 6, 11–13] that explain the topic of concept drift, present different taxonomies of the respective methods, and provide insight into application areas.

This paper provides a promising contribution to address the stationarity challenge in SHM. The primary goal of this work is to investigate the CDD methods and to introduce an innovative approach for the selection of a suitable method.

The remainder of this paper is organized as follows. In Section II there is a brief overview of the basics of concept drift. The methodology used in this research is described in Section III and Section IV presents and discusses the results. Section V describes the proposed approach of method selection. In the last section, the findings are elaborated, and future directions pointed out.

CONCEPT DRIFT DETECTION

Figure 1 shows the process flow of CDD. After feature extraction, the process starts simultaneously with the conventional ML model. CDD monitors the data distributions and detects changes, and concept drift adaptation (CDA) adjusts the ML model to address the effects of drifting data. A distinction is made between active and passive approaches to CDA. In passive mode, the model updates regularly, regardless of drift occurrence, maintaining high accuracy but with high computational costs. Active approaches, on the other hand, adjust the model only when a drift is detected, making

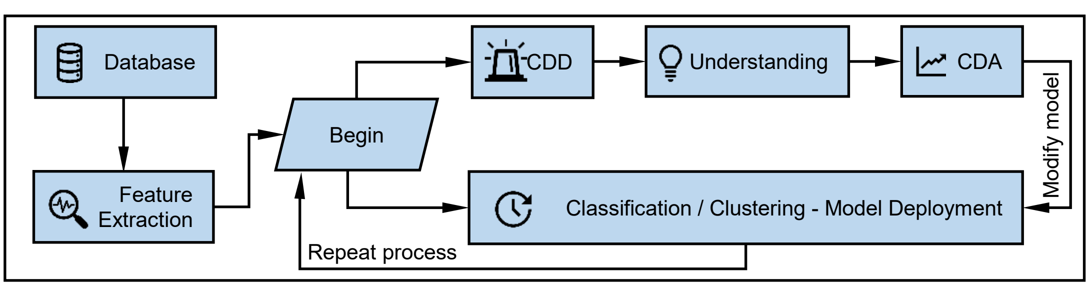

Figure 1. Process flow of CDD.

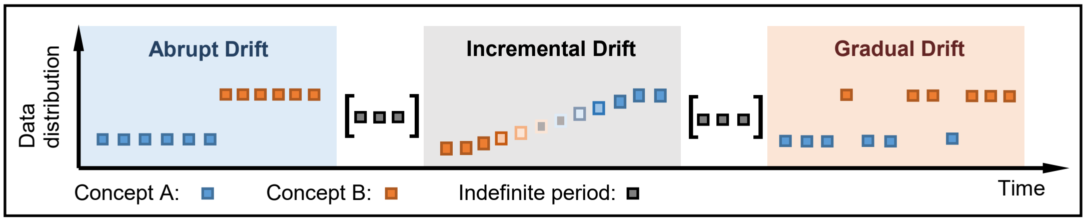

Figure 2. Types of concept drift.

them more resource efficient. However, a drawback is that undetected drift can lead to decreased prediction quality over time. In addition, some methods also provide drift understanding. This process can then start again until the next concept drift is detected.

Concept drift occurs in different forms (see Figure 2). The differentiation between different types of concept drift is particularly evident in the relationship between input data and target variable, but the speed of drift occurrence is also a distinguishing feature. If the change from one concept to the next occurs rapidly, *abrupt drift* is present. Processes that take several time steps until a concept A transfers to a concept B are called *gradual drift* or *incremental drift*. Unlike incremental drift, which usually exhibits monotonic behavior in the direction of the new concept, gradual drift overlaps both concepts [13]. The change occurs as concept A is successively replaced by the subsequent concept. In general, abrupt drifts are easy to detect because they mark an incisive point in the course of the data stream. In contrast, incremental processes present a greater challenge. The change per time step can be so minimal that it can be mistaken for natural noise or measurement errors. Only when a longer time period is considered does the continuous variation become apparent [14]. The complexity of detecting gradual drift is heavily dependent on the specific characteristics of the data stream.

Several reviews such as [3, 13, 15] provide a broad overview of the full range of applicable methods for detecting concept drift in supervised and unsupervised scenarios. Table I outlines the study's methods, while a more detailed explanation of the method selection is provided in the subsequent section.

TABLE I. OVERVIEW OF THE SELECTED METHODS

Method	Label availability	Drift type detection	Significant properties
EDDM [16]	Supervised	Gradual, (abrupt), incremental drift	Better results for gradual drifts
$FHDDMS_{add}$ [17]	Supervised	Gradual, abrupt, incremental drift	High drift detection delay
MDDM [18]	Supervised	Gradual, abrupt, incremental drift	Fast Detection of Concept Drift
EFDT [19]	Supervised	Gradual, (abrupt) drift	Fast learning rate, slow drift recovery
CluStream [20]	Unsupervised	Gradual, abrupt drift	Determinacy of parameters
ClusTree [21]	Unsupervised	Gradual, abrupt drift	Parameter-free method
DenStream [22]	Unsupervised	Abrupt drift	No determinacy of parameters
StreamKM++[23]	Unsupervised	Gradual, (abrupt) drift	Low runtime, fixed number of clusters
D-Stream [24]	Unsupervised	-	High cluster quality, no CDD

METHODOLOGY

To generate a comprehensive overview of the existing CDD methods, a systematic literature review is carried out. The procedure can be described as follows:

1. Selection of databases: The literature search is based on the databases IEEEXplore, ACM Digital Library, Springer Link, Semantic Scholar, and ScienceDirect.
2. Pre-selection by keywords: The pre-selection of the results (preliminary screening) is done using keywords that indicate that either a new method for dealing with concept drift or an overview of the field of concept drift is presented.
3. Result filtering: In this step, the results are filtered either temporally (over a period of the last 20 years) or based on their influence (in this case defined by a publication in high quality journals and conferences or a particularly high number of citations).

At the time of writing this paper, a total of 342 methods for CDD have been identified, following the methodology described above. The selection of methods for the experiments is based on several decision criteria. First, there is a temporal prioritization. Methods that represent an optimization to another method were considered in the selection in so far as results from comparative studies were included. A further indication can be seen in the citation reliability, which is also recorded for all methods. Based on this decision criteria, nine CDD methods are evaluated in ten experiments with respect to different types of concept drift. The experiments are conducted using the open-source evaluation platform (Massive Online Analysis) MOA that offers a range of tools for the generation of data streams and enables the synthetic modeling of drifts [25].

Examinations are conducted on three synthetic and three real datasets. Synthetic data are chosen for its suitability in investigating different types of drift in a systematic manner. Simultaneously, the inclusion of real datasets is essential to explore the performance of methods in unknown or more intricate systems. The SEA [26], Agrawal [27] and Mixed [28] generators were used in MOA [25] to generate synthetic datasets with known drift behavior. The datasets each contain 100,000,000 instances to adequately represent the behavior of data streams. The Forest covertype dataset, Poker Hand dataset and Electricity market dataset were used as real datasets.

The central metric in the study of classification tasks in the supervised field is the accuracy of the learning algorithm. In the evaluation of data streams, the time aspect is another crucial factor for the feasibility of the method. A shorter evaluation time expands the range of potential applications. Ultimately, it is necessary to address the model's costs by considering efficient utilization of computational capacity.

For the unsupervised case, where the ground truth is not always available, a new approach is required. Since each metric has its individual advantages and disadvantages, a combined average of the examined metrics CMM (Cluster Mapping Measures), Purity, F1-P (F1-Precision), F1-R (F1-Recall) and the Silhouette coefficient is formed [29]. SSQ (Sum of Square Distances) is considered separately due to the different scaling:

$$Q_{AVG} = \frac{CMM+Purity+F1P+F1R+SilCoeff}{5} \quad (2)$$

RESULTS AND DISCUSSION

The results of the synthetic and real test series are listed in Table II. For each supervised method the evaluation metrics accuracy post drift, evaluation time and model cost are considered. Furthermore, for the unsupervised methods, the evaluation is done according to the quality metrics SSQ and QAVG or Silhouette coefficient in the real use case. Readers who wish to obtain full results for all experiments described in this paper may contact the corresponding authors for access to the data.

In the context of synthetic data streams used to generate known drift events, and real datasets representing unknown drift behavior, the following conclusions can be drawn:

Without any form of CDD, the accuracy of the model drops by an average of 20%. Such a loss of accuracy is unacceptable in most cases and would render the model unusable, further emphasizing the need for CDD. Furthermore, EFDT performs comparatively poorly on average, although it has the highest accuracy values due to a better base classifier. The deficiencies in the detection of abrupt and incremental drifts lead to an average accuracy of 66.49%. EDDM, $FHDDMS_{add}$, and MDDM-E, are close in terms of accuracy, time, and storage complexity, with EDDM performing best here. EFDT performs worst in the context of time and memory complexity as can be seen in Table II. The method is characterized by its exponential nature, which makes it unsuitable for large streaming scenarios. For the unsupervised case, D-Stream is superior to the other methods in all three evaluation metrics, although this is not due to a high adaptability to concept drift, but to the general high cluster quality. Furthermore, the methods CluStream and ClusTree show only marginal differences in their performance. StreamKM++ performs worst overall, which can be explained by the lack of adaptability in gradual drifts and a similar performance in real world application scenarios. In summary, ClusTree is especially suitable for fast data streams. StreamKM++ and DenStream are inferior to the other methods both in accuracy and computation time. D-Stream has the highest quality and comparatively low requirements, but does not work as a CDD method like, for example, CluStream.

TABLE II. EVALUATION RESULTS FOR THE INVESTIGATED METHODS (THE TWO BEST ARE MARKED IN BOLD).

	Accuracy post drift (mean) [%]		**Evaluation time [cpu-sec.]**		**Model cost [RAM-h]**	
Method (supervised)	Synthetic datasets	Real datasets	Synthetic datasets	Real datasets	Synthetic datasets	Real datasets
No drift detection	58,83	64,49	**530,45**	7,98	**1,87E-7**	**7,34E-8**
EDDM	82,03	**82,80**	569,55	**7,89**	1,14E-6	6,40E-8
$FHDDMS_{add}$	**82,52**	80,33	**565,63**	**7,91**	1,16E-6	6,74E-8
MDDM-E	**82,27**	**80,95**	585,95	7,96	**1,32E-6**	**6,93E-8**
EFDT	66,49	80,65	762,56	25,71	2,21E-3	1,83E-5
	Q_AVG (mean)		**SilCoeff**		**SSQ**	
Method (unsupervised)	Synthetic datasets	Real datasets	Synthetic datasets	Real datasets	Synthetic datasets	Real datasets
CluStream	0,64	-	0,63	**0,82**	1,45E16	-
ClusTree	**0,65**	-	0,63	0,77	**1,42E16**	-
DenStream	0,49	-	**0,79**	0,58	**1,42E16**	-
D-Stream	**0,81**	-	**1,00**	**1,00**	**8,95E15**	-
StreamKM++	**0,65**	-	0,57	0,64	1,60E16	-

PROPOSED APPROACH

Based on these findings and an in-depth literature research of more than 340 methods, it can be concluded that the selection of an appropriate method is not trivial given the vast range of options and specifications, leading to the frequent selection of suboptimal methods. Moreover, although it is evident that the detection of concept drift has a crucial contribution to the maintenance of ML models, the field is very specific and therefore rarely applied in many application areas such as SHM or industrial system monitoring. Therefore, we propose a new and innovative approach that addresses method selection and can assist researchers and practitioners from different domains in selecting an appropriate method for their specific applications.

Figure 3 presents the approach outlined here. The basis of this approach is a central database or an IoT system to manage the data. A graphical user interface (GUI) is coupled to this, which represents the human-machine interface (HMI). This is where individual process parameters and the constraints of the applications are submitted to an agent. This includes information such as label availability and data acquisition, which have a significant impact on the selection of methods and enable a basic differentiation between supervised, unsupervised, and semi-supervised methods. Information from experts in the respective domain regarding commonly occurring types of anomalies, limitations in terms of resource availability and preferences in the architecture of the monitoring (passive/ active monitoring of concept drift) are central issues in the selection of a suitable method. The agent takes these preferences into account and gives them as input to an AI-based selection algorithm. This is trained using various applications to predict, which of the available methods is best suited for the input data. After evaluation on a training data set, the algorithm gives the agent a selection of one or a few methods that have performed best. The user can then select and deploy these via the GUI. Within this framework, more information relevant to the user can be

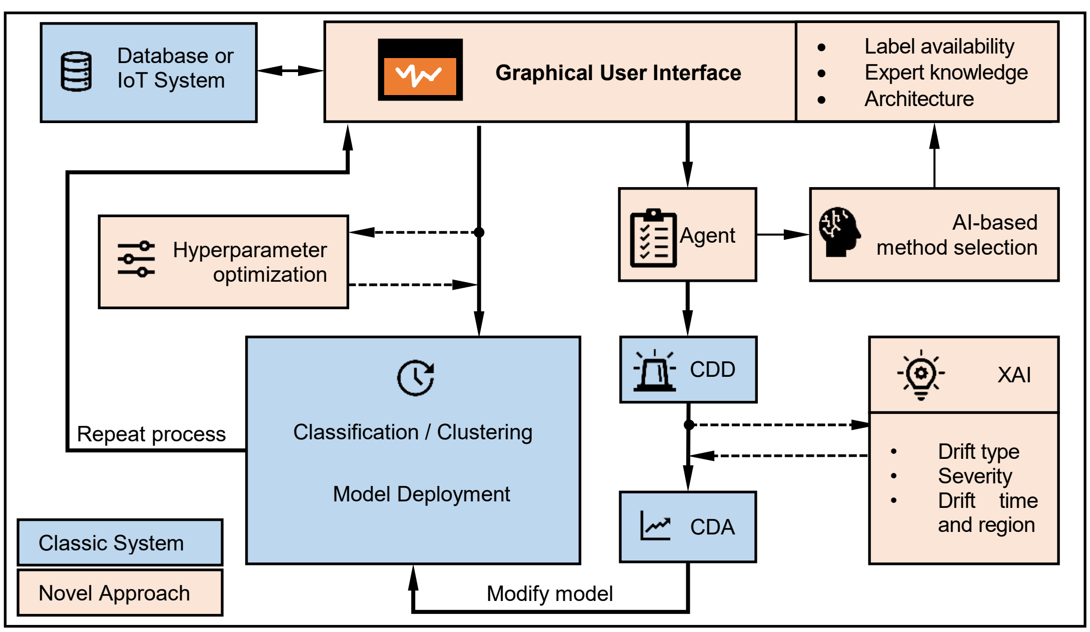

Figure 3. Proposed Approach.

provided. For example, the entire training process can be automated with a subsequent hyperparameter optimization phase. Furthermore, the quality and performance of the methodology during the entire life cycle is monitored and an understanding of the retraining can be created using Explainable AI (XAI) when detecting concept drift. Information regarding type of drift, severity and exact timing of the drift occurrence is highly relevant in this context. By considering the characteristics of the data stream and the requirements of the application, this approach provides a systematic and efficient way to select the most appropriate CDD method.

CONCLUSION: FINDINGS AND FUTURE DIRECTIONS

This paper addresses the treatment of concept drift in SHM data streams. For this purpose, a systematic literature search was conducted. After identifying the specific challenges and characteristics for the present circumstances, a selection of nine methods was made. To analyze the quality and efficiency of the respective methods, the open-source platform MOA was used, and a study was conducted in ten experiments, which investigated the behavior of the methods on specific drift patterns. The results have shown that an implementation of a concept drift method allows an average improvement of the accuracy by 20% and thus it could be verified that the necessity of the adaptation is given. Furthermore, it was found that none of the methods was far superior to the competitors in all scenarios. Depending on the drift type, differences in performance could be found. Within this work it becomes obvious that the process of finding a suiting method is far from trivial and requires special attention. For this reason, an innovative, holistic approach is presented, which integrates the selection process into the life cycle of the ML model and thus enables the user to achieve a symbiosis of CDD and a specific use case like SHM. Just as conventional damage detection techniques are gradually replaced by state-of-the-art smart monitoring and decision-making solutions, these ML techniques need to be updated or completely retrained over time.

The methods evaluated here are not the most representative but compare different levels of complexity and will be continuously extended by current methods in the next step. Especially the investigation of the unsupervised methods requires further research, for example on the influence of parameters deviating from the recommended settings and on the effect of stream speed. Furthermore, the implementation and evaluation of the proposed approach for method selection in real SHM use cases is targeted.

REFERENCES

[1] Malekloo, A., Ozer, E., AlHamaydeh, M., and Girolami, M. 2022. "Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights," *Structural Health Monitoring*, 21(4):1906–1955.

[2] Muin, S., and Mosalam, K. M. 2021. "Structural Health Monitoring Using Machine Learning and Cumulative Absolute Velocity Features," *Applied Sciences*, 11(12).

[3] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. 2019. "Learning under Concept Drift: A Review," *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363.

[4] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. 2014. "A Survey on Concept Drift Adaptation," *ACM Comput. Surv.*, 46(4).

[5] Schlimmer, J. C., and Granger, R. H. 1986. "Incremental learning from noisy data," *Machine Learning*, 1(3):317–354.

[6] Wares, S., Isaacs, J., and Elyan, E. 2019. “Data stream mining: methods and challenges for handling concept drift,” *SN Applied Sciences*, 1(11):1412.
[7] Žliobaitė, I. 2010. “Learning under Concept Drift: an Overview,” *CoRR*, abs/1010.4784.
[8] Bifet, A., and Gavaldà, R. “Learning from Time-Changing Data with Adaptive Windowing,” *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*:443–448.
[9] Huang, G.-B., Liang, N., Rong, H.-J., Saratchandran, P., and Sundararajan, N. 2005. “On-Line Sequential Extreme Learning Machine,” 2005:232–237.
[10] Jia, S. 2020. “A VFDT algorithm optimization and application thereof in data stream classification,” *Journal of Physics: Conference Series*, 1629(1):12027.
[11] Iwashita, A. S., and Papa, J. P. 2019. “An Overview on Concept Drift Learning,” *IEEE Access*, 7:1532–1547.
[12] Ditzler, G., Roveri, M., Alippi, C., and Polikar, R. 2015. “Learning in Nonstationary Environments: A Survey,” *IEEE Computational Intelligence Magazine*, 10(4):12–25.
[13] Agrahari, S., and Singh, A. K. 2022. “Concept Drift Detection in Data Stream Mining : A literature review,” *Journal of King Saud University - Computer and Information Sciences*, 34(10, Part B):9523–9540.
[14] Hoens, T. R., Polikar, R., and Chawla, N. V. 2012. “Learning from streaming data with concept drift and imbalance: an overview,” *Progress in Artificial Intelligence*, 1(1):89–101.
[15] Gerz, F., Bastürk, T. R., Kirchhoff, J., Denker, J., Al-Shrouf, L., and Jelali, M. 2022. “A Comparative Study and a New Industrial Platform for Decentralized Anomaly Detection Using Machine Learning Algorithms,” *International Joint Conference*:1–8.
[16] Baena-García, M., Del Campo-Ávila, J., Bifet, A., Gavaldà, R., and Morales-Bueno, R. 2006. “Early Drift Detection Method,”.
[17] Pesaranghader, A., Viktor, H., and Paquet, E. 2018. “Reservoir of diverse adaptive learners and stacking fast hoeffding drift detection methods for evolving data streams,” *Machine Learning*, 107(11):1711–1743.
[18] Pesaranghader, A., Viktor, H., and Paquet, E. 2017. “McDiarmid Drift Detection Methods for Evolving Data Streams,”.
[19] Manapragada, C., Webb, G. I., and Salehi, M. 2018. “Extremely Fast Decision Tree,” *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* Association for Computing Machinery:1953–1962.
[20] Charu C. Aggarwal, Philip S. Yu, Jiawei Han, and Jianyong Wang. 2003. “- A Framework for Clustering Evolving Data Streams,” *Proceedings 2003 VLDB Conference,* edited by Johann-Christoph Freytag, et al., Morgan Kaufmann, San Francisco:81–92.
[21] Kranen, P., Assent, I., Baldauf, C., and Seidl, T. 2011. “The ClusTree: indexing micro-clusters for anytime stream mining,” *Knowledge and Information Systems*, 29(2):249–272.
[22] Cao, F., Estert, M., Qian, W., and Zhou, A. 2006. “Density-Based Clustering over an Evolving Data Stream with Noise,” *Proceedings of the Sixth SIAM International Conference on Data Mining,* Society for Industrial and Applied Mathematics:328–339.
[23] Ackermann, M. R., Märtens, M., Raupach, C., Swierkot, K., Lammersen, C., and Sohler, C. 2012. “StreamKM++: A Clustering Algorithm for Data Streams,” *ACM J. Exp. Algorithmics*, 17.
[24] Chen, Y., and Tu, L. 2007. “Density-Based Clustering for Real-Time Stream Data,” *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* Association for Computing Machinery:133–142.
[25] Bifet, A., Holmes, G., Pfahringer, B., Kranen, P., Kremer, H., Jansen, T., and Seidl, T. 2010. “MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering,” *Proceedings of the First Workshop on Applications of Pattern Analysis,* PMLR, 11:44–50.
[26] Street, W. N., and Kim, Y. 2001. “A streaming ensemble algorithm (SEA) for large-scale classification,” *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining / KDD-2001,* ACM Press:377–382.
[27] Agrawal, R., Imielinski, T., and Swami, A. 1993. “Database mining: a performance perspective,” *IEEE Transactions on Knowledge and Data Engineering*, 5(6):914–925.
[28] Gama, J., Medas, P., Castillo, G., and Rodrigues, P. “Learning with Drift Detection,” *Hutchison, Kanade et al. (Hg.) – Advances in Artificial Intelligence*, 3171:286–295.
[29] Jaime Andrés Merino. 2015. “Streaming Data Clustering in MOA using the Leader Algorithm,”.