# Ensembles of Heterogeneous Concept Drift Detectors - Experimental Study

Michal Woźniak, Pawel Ksieniewicz, Boguslaw Cyganek, Krzysztof Walkowiak

# Ensembles of Heterogeneous Concept Drift Detectors - Experimental Study

Michał Woźniak[1], Paweł Ksieniewicz[1], Bogusław Cyganek[2], and Krzysztof Walkowiak[1]

[1] Department of Systems and Computer Networks, Faculty of Electronics
Wrocław University of Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland.
e-mail: [michal.wozniak, pawel.ksieniewicz,
krzysztof.walkowiak]@pwr.edu.pl

[2] AGH University of Science and Technology Al. Mickiewicza 30, 30-059 Kraków, Poland e-mail: cyganek@agh.edu.pl

**Abstract.** For the contemporary enterprises, possibility of appropriate business decision making on the basis of the knowledge hidden in stored data is the critical success factor. Therefore, the decision support software should take into consideration that data usually comes continuously in the form of so-called *data stream*, but most of the traditional data analysis methods are not ready to efficiently analyze fast growing amount of the stored records. Additionally, one should also consider phenomenon appearing in data stream called *concept drift*, which means that the parameters of an using model are changing, what could dramatically decrease the analytical model quality. This work is focusing on the classification task, which is very popular in many practical cases as fraud detection, network security, or medical diagnosis. We propose how to detect the changes in the data stream using combined concept drift detection model. The experimental evaluations confirm its pretty good quality, what encourage us to use it in practical applications.

**Keywords:** data stream, concept drift, pattern classification, drift detector

## 1 Introduction

The analysis of huge volumes and fast arriving data is recently the focus of intense research, because such methods could build a competitive advantage of a given company. One of the useful approach is the data stream classification, which is employed to solve problems related to discovery client preference changes, spam filtering, fraud detection, and medical diagnosis to enumerate only a few.

However, most of the traditional classifier design methods do not take into consideration that:

- The statistical dependencies between the observations of the given objects and their classifications could change.

– Data can arrive so quick that labeling all records is impossible.

This section focuses on the first problem called *concept drift* [24] and it comes in many forms, depending on the type of change. Appearance of concept drift may potentially cause a significant accuracy deterioration of an exploiting classifier. Therefore, developing positive methods which are able to effectively deal with this phenomena has become an increasing issue. In general, the following approaches may be considered to deal with the above problem.

– Frequently rebuilding a model if new data becomes available. It is very expensive and impossible from a practical point of view, especially when the concept drift occurs rapidly.
– Detecting concept changes in new data, and if these changes are *sufficiently* significant then rebuilding the classifier.
– Adopting an incremental learning algorithm for the classification model.

Let's firstly characterize shortly the probabilistic model of the classification task [4]. It assumes that attributes $x \in \mathcal{X} \subseteq \mathcal{R}^d$ and class label $j \in \mathcal{M} = \{1, 2, ..., M\}$ are the observed values of a pair of random variables $(\mathbf{X}, \mathbf{J})$. Their distribution is characterized by *prior* probability

$$p_j = P(\mathbf{J} = j),\ j \in \mathcal{M} \tag{1}$$

and conditional probability density functions [3]

$$f_j(x) = f(x|j),\ x \in \mathcal{X},\ j \in \mathcal{M}. \tag{2}$$

The classifier $\Psi$ maps feature space $\mathcal{X}$ to the set of the class labels $\mathcal{M}$

$$\Psi\ :\ \mathcal{X} \to \mathcal{M}. \tag{3}$$

The optimal Bayes classifier $\Psi^*$ minimizes probability of missclassification according to the following classification rule [4]:

$$\Psi^*(x) = i \text{ if } p_i(x) = \max_{k \in \mathcal{M}} p_k(x), \tag{4}$$

where $p_j(x)$ stands for *posterior* probability

$$p_i(x) = \frac{p_i f_i(x)}{f(x)} \tag{5}$$

and $f(x)$ is unconditionally density function.

$$f(x) = \sum_{k=1}^{M} p_k\ f_k(x) \tag{6}$$

[3] We assume continuous attributes, but for discrete ones we have to take into consideration corresponding conditional probabilities $p(x|j)$

[4] We assume so-called 0-1 loss function. For different loss function the optimal algorithm minimizes so-called overall risk [4].

Backing to the concept drift problem we may distinguished two types of drifts according its influence into probabilistic characteristics of the classification task [5]:

- *virtual concept drift* means that the changes do not have any impact on decision boundaries (some works report that they do not have an impact on *posterior* probability, but it is disputable), but they change unconditionally density functions [23].
- *real concept drift* means that the changes have an impact on decision boundaries, i.e., on *posterior* probabilities [20, 24].

Considering the classification task, the real concept drift is the most important, but detecting the virtual one could be useful as well, because in the case if we are able not only to detect the drift, but also distinguish between virtual and real concept drift, then we may decide if classifier's model rebuilding is necessary or not. Another taxonomy of concept drift bases on its impetuosity. We may distinguish:

- Slow changes (*gradual or incremental drift*). For the gradual drift for a given period of time examples from different models could appear in the stream concurrently, while for incremental drift the model's parameters are changing smoothly.
- Rapid changes (*sudden drift*, *concept shift*).

A plethora of solutions have been proposed how to deal with this phenomena. Basically, we may divide these algorithms into four main groups:

1. Online learners [25]
2. Instance based solutions (also called sliding window based solutions)
3. Ensemble approaches [12, 14, 17]
4. Drift detection algorithms

In this work we will focus on the drift detectors which are responsible for determining whether two successive data chunks were generated by the same distribution [10]. In the case the the change is detected the decision about collecting new data to rebuild new model could be made.

## 2 Concept drift detection

As we mentioned above the appearance of concept drift may potentially cause a significant accuracy deterioration of an exploiting classifier [16], what is shown in Fig. 1.

Concept drift detector is an algorithm, which on the basis of incoming information about new examples and their correct classification or a classifier's performance (as accuracy) can return information that data stream distributions are changing. The currently used detectors return only signal about drift detection, which usually requires a quick classifier's model updating or that warning

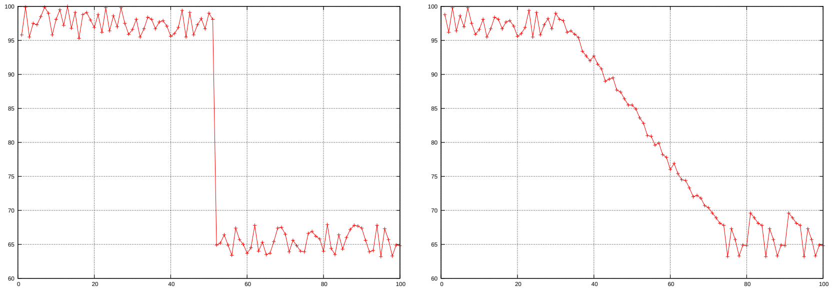

**Fig. 1.** Exemplary deterioration of classifier accuracy for sudden drift (left) and incremental one (right).

level is achieved, which is usually treated as a moment that new learning set should be gathered (i.e., all new incoming examples should be labeled). The new learning set is used to update the model is drift is detected. The idea of drift detection is presented in Fig. 2

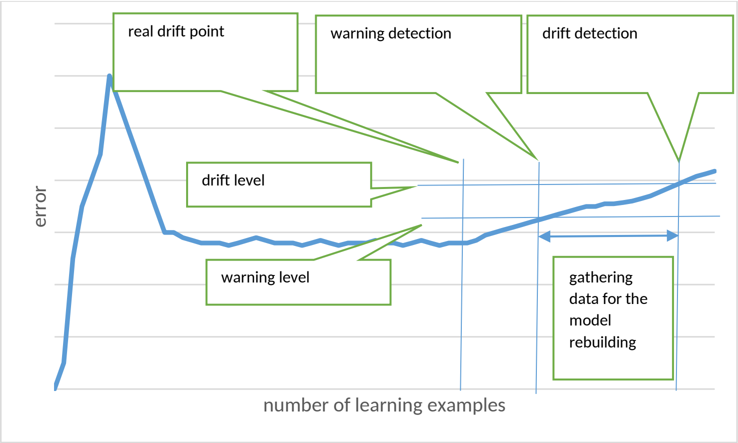

**Fig. 2.** Idea of drift detection.

The drift detection could be recognized as the simple classification task, but from practical point of view detectors do not use the classical classification model. The detection is hard, because on the one hand we require quick drift detection to quickly replace outdated model and to reduce so-called restoration time. On the other hand we do not accept false alarms [9], i.e., when detector returns that change appears, but it is not true. Therefore to measure performance of concept drift detectors the following metrics are usually used:

- Number of correct detected drifts.

- Number of false alarms.
- Time between real drift appearance and its detection.

In some works, as [2], aggregated measures, which take into consideration the mentioned above metrics, are proposed, but we decided not to use them because using aggregated measures do not allow to precisely analyse behavior of the considered detectors.

## 3 Description of the chosen drift detectors

Let us shortly describe selected methods of drift detection, which later will be used to produce combined detectors.

### 3.1 DDM(*Drift Detection Method*)

This algorithm [6] is analyzing the changes in the probability distribution of the successive examples for different classes. On the basis of the mentioned above analysis DDM estimates classifier error, which (assuming the convergency of the classifier training method) has to decrease, what means the probability distribution does not change [19]. If the classifier error is increasing according to the number of training examples then this observation suggests a change of probability distribution and the current model should be rebuilt. DDM estimates classification error, its standard deviation and stores their minimal values. If estimated error is greater than stored minimal value of error and two times its standard deviation then DDM returns signal that the warning level is achieved. In the case if estimated error is greater than stored minimal value of error and three times its standard deviation then DDM returns signal that drift is detected.

### 3.2 CUSUM (*CUmulative SUM*)

CUSUM [18] detects a change of a given parameter value of a probability distribution and indicated when the change is significant. As the parameter the expected value of the classification error could be considered, which may be estimated on the basis of labels of incoming objects from data stream. The detection condition looks as follows

$$g_t = max(0, g_{t-1} + \epsilon_t - \xi) > treshold \tag{7}$$

where $g_0 = 0$ and $\epsilon_t$ stands for observed value of a parameter in time $t$ (e.g., mentioned above classification error). $\xi$ describes rate of change. Te value of *treshold* is set by an user and it is responsible for detector's sensitivity. Its low value allows to detect drift quickly, but then a pretty high number of false alarms could appear [2, 7].

### 3.3 Test PageHinkley

It is modification of the CUSUM algorithm, where the cumulative difference between observed classifier error and its average is taken into consideration [21].

### 3.4 Detectors based on Hoeffding's and McDiarmid's inequalities

The interesting drift detectors based on non-parametric estimation of classifier error employing Hoeffding's and McDiarmid's inequalities were proposed in [3].

## 4 Combined concept drift detectors

Let's assume that we have a pool of $n$ drift detectors

$$\Pi = \{D_1, D_2, ..., D_n\} \tag{8}$$

Each of them returns signal that warning level is achiever or concept drift is detected, i.e.,

$$D_i = \begin{cases} 0 \; if \; drift \; is \; not \; detected \\ 1 \; if \; warning \; level \; is \; achieved \\ 2 \; if \; drift \; is \; detected \end{cases} \tag{9}$$

As yet not so many papers deal with combined drift detectors. Bifet et al. [2] proposed the simple combination rules based on the appearance of drift once ignoring signals about warning level. The idea of combined drift detector is presented in Fig.4.

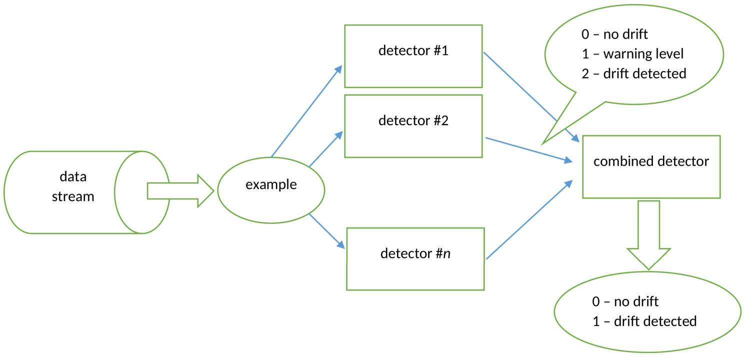

**Fig. 3.** Idea of combined drift detector.

Let's present three combination rules which allow to combine the concept drift detector outputs. Because this work focuses on the ensemble of heterogeneous detectors, therefore only the combination rules using drift appearance signal are taken into consideration.

### 4.1 ALO (*At Least One detects drift*)

Committee of detectors makes a decision about drift if at least one individual detector returns decision that drift appears (not all of the detectors can return warning signal).

$$ALO(\Pi, x) = \begin{cases} 0 \sum_{i=1}^{n} [D_i(x) = 2] \\ 1 \sum_{i=1}^{n} [D_i(x) = 0] \end{cases} \tag{10}$$

where [] denotes Iverson bracket.

### 4.2 ALHD (*At Least Half of the detectors detect Drift*)

Committee of detectors makes a decision about drift if at half of individual detectors return decisions that drift appears.

$$ALHWD(\Pi, x) = \begin{cases} 0 \sum_{i=1}^{n} [D_i(x) = 2] < \frac{n}{2} \\ 1 \sum_{i=1}^{n} [D_i(x) = 2] \geqslant \frac{n}{2} \end{cases} \tag{11}$$

### 4.3 AD (*All detectors detect Drift*)

Committee of detectors makes a decision about drift if each individual detector returns decisions that drift appears.

$$ALHWD(\Pi, x) = \begin{cases} 0 \sum_{i=1}^{n} [D_i(x) = 2] < n \\ 1 \sum_{i=1}^{n} [D_i(x) = 2] \geqslant n \end{cases} \tag{12}$$

Let's notice that proposed combined detectors are *de facto* classifier ensembles [26], which use deterministic (untrained) combination rules. They do not take into consideration any additional information as about individual drift detector's qualities. In this work, we also do not focus on the very important problem how choose the valuable pool of individual detectors. For classifier ensemble such a process (called ensemble selection or ensemble pruning) uses diversity measure [15], but for the combined detectors the measures have been using, thus far is impossible, because of different nature of the decision task.

## 5 Experimental research

### 5.1 Goals

The main objective of the experimental study was evaluating the proposed combined concept drift detectors and their comparison with the well-known simple methods. To ensure the appropriate diversity of the pool of detectors we decided to produce an ensemble on the basis of five detectors employing different models presented in the previous section. For each experiment we estimated detector sensitivity, number of false alarms and computational complexity of the model, i.e., commutative running time.

### 5.2 Set-up

We used the following models of individual detectors:

- DDM with window includes 30 examples.
- HDDM_A - detector based on McDiarmid's inequality analysis with the following parameters: drift confidence 0,001, warning confidence 0,005, and one-side t-test.
- HDDM_W - detector based on Hoeffding's inequality analysis with the following parameters: drift confidence 0,001, warning confidence 0,005, and one-side t-test.
- Cusum with window includes 30 examples, $\delta = 0,005$ and $lambda = 50$
- Page-Hinkley test with window includes 30 examples, $\delta = 0,005$, $lambda = 50$, and $\alpha = 1$

As we mentioned above we use the individual detectors to build three combined ones based on the ALO, ALHD, and AD combination rules.

All experiments were carried out using MOA (*Massive Online Analysis*)[5] and our own software written in Java according to MOA's requirements [1].

For each experiment 3 computer generated data streams were used. Each of them consists of 10 000 examples:

- Data stream with sudden drift appearing after each 5.000 examples.
- Data stream with gradual drift, where 2 concept appear.
- Data stream without drift.

The stationary data stream (without drift) was chosen because we would like to evaluate the sensitivity of the detector, i.e., number of false alarms.

### 5.3 Results

The results of experiment were presented in Tab.1-3 [6]

**Table 1.** Results of experiment for data stream with sudden drift

| detector | no. of real drifts | no. of corrected detected drifts | average detection delay | no. of false detections | time |
|---|---|---|---|---|---|
| DDM | 20 | 20 | 71,45 | 0 | 1,47 |
| HDDM_A | 20 | 19 | 19 | 1 | 1,79 |
| HDDM_W | 20 | 20 | 11,65 | 0 | 1,71 |
| PageHinkley | 20 | 1 | 102 | 0 | 1,61 |
| CUSUM | 20 | 20 | 141,95 | 0 | 1,59 |
| ALO | 20 | 0 | - | 78 | 4,93 |
| ALHD | 20 | 1 | 102 | 0 | 4,76 |
| AD | 20 | 20 | 29,8 | 6 | 4,85 |

[5] http://moa.cms.waikato.ac.nz/

[6] The detailed results of the experiments could be found `https://drive.google.com/drive/u/0/folders/0B8ja_TIQel7KbnJMblJJUzltNzQ`

**Table 2.** Results of experiment for data stream with gradual drift

| detector | no. of real drifts | no. of corrected detected drifts | average detection delay | no. of false detections | time |
|---|---|---|---|---|---|
| DDM | 1 | 1 | 235 | 2 | 1,44 |
| HDDM_A | 1 | 1 | 310 | 1 | 1,65 |
| HDDM_W | 1 | 1 | 242 | 1 | 1,62 |
| PageHinkley | 1 | 1 | 340 | 1 | 1,59 |
| CUSUM | 1 | 1 | 346 | 1 | 1,62 |
| ALO | 1 | 1 | 235 | 10 | 4,89 |
| ALHD | 1 | 1 | 346 | 1 | 4,86 |
| AD | 1 | 1 | 299 | 2 | 4,95 |

**Table 3.** Results of experiment for data stream without drift

| detector | no. of real drifts | no. of corrected detected drifts | average detection delay | no. of false detections | time |
|---|---|---|---|---|---|
| DDM | 0 | 0 | - | 0 | 1,4 |
| HDDM_A | 0 | 0 | - | 0 | 1,65 |
| HDDM_W | 0 | 0 | - | 0 | 1,59 |
| PageHinkley | 0 | 0 | - | 0 | 1,53 |
| CUSUM | 0 | 0 | - | 0 | 1,61 |
| ALO | 0 | 0 | - | 0 | 7,79 |
| ALHD | 0 | 0 | - | 0 | 4,84 |
| AD | 0 | 0 | - | 0 | 4,92 |

### 5.4 Discussion

Firstly we have to emphasize that we realize that the scope of the experiments was limited therefore drawing the general conclusions is very risky. For each experiments the detector based on Hoeffding's inequality (HDDM_W) [3] outperformed other models. The combined detectors have not behaved well for sudden and frequent drift (see Tab.1), because some of them (as ALO) has to low sensitivity or so high sensitivity (as for AD) caused a high number of false alarms. The computation time of AD detector was similar as in the case of HDDM_A or HDDM_W, but as we mentioned before the number of false alarms was not acceptable. For gradual drift the combined detectors ALHD and AD behaved similar as individual detectors. For stationary stream they have not presented so high sensitivity (similar as individual detectors). Probably, not so impressive results of combined detectors have been caused by the fact that we nominated the the individual models arbitrary and we did not check how adding or removing detector from the pool could impact on the combined detector quality. Additionally, HDDM_W and HDDM_A detectors strongly outperformed other models, what could cause that the decision of the detector ensemble was the same as the dominant models. The similar observation has been reported for the classifier ensemble using deterministic combination rules. On the basis of the experiment we are not able to say expressly if the combined concept drift detectors are promising direction. Nevertheless, we decided to continue the works on such models, especially for a pool of homogeneous detectors and method which are able to prune the detector ensemble. We have also to notice the main drawback of the proposed models, which are more complex than single ones. On the other hand we have to notice that using the proposed method of parallel interconnection is easy to parallelize and could be run in a distributed computing environment.

## 6 Final remarks

In this work three simple combination rules for combined concept drift detectors were discussed and evaluated on the basis of the computer experiments for different data streams. They seem to be an interesting proposition to solve the problem of concept drift detection, nevertheless the results of experiments do not confirm their high quality, but as we mentioned above it is probably caused by very naive choice of the individual models of the detector ensemble.

It is worth noticing that we assume the continue access to class labels. Unfortunately, from the practical point of view it is hard to be granted that labels are always and immediately available, e.g., for credit approval task the true label is usually available ca. 2 years after the decision, then such labeled example could be worthless as come from outdated model. Therefore, during constructing the concept drift detectors we have to take into consideration the cost of data labeling, which is usually passed over. It seems to be very interesting to design detectors on the basis of a partially labeled set of examples (called *active learning*) [8] or unlabeled examples. Unfortunately, unsupervised drift detectors can detect the virtual drift only, but it is easy to show that without access to class labels (then we can analyse the unconditionally probability distribution only $f(x)$) the real drift could be undetected [22].

Let's propose the future research directions related to the combined concept drift detectors:

- Developing methods how to choose individual detectors to a committee, maybe dedicated diversity measure should be proposed. In this work the diversity of the detectors was ensured by choosing different detector's models, but we may also use the same model of detector but with different parameters, e.g., using different drift confidences and warning confidences for detector based on McDiarmid's inequality.
- Proposing semi-supervised and unsupervised methods of combined detector training.
- Proposing trained combination rule to establish the final decision of the combined detector and to fully exploit the strengths of the individual detectors.
- Developing the combined local drift detectors, probably employing the clustering and selection approach [11, 13], because many changes have the local nature and touch the selected area of the feature space or selected features only.
- Employing other interconnections of detectors (in this work the parallel architecture was considered only) including serial one.

## Acknowledgements

This work was supported by the statutory funds of the Department of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Science and Technology and by the Polish National Science Centre under the grant no. DEC-2013/09/B/ST6/02264. This work was also supported by the AGH Statutory Funds no. 11.11.230.017. All computer experiments were carried out using computer equipment sponsored by ENGINE project(http://engine.pwr.edu.pl/).

## References

1. Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. Moa: Massive online analysis. *J. Mach. Learn. Res.*, 11:1601–1604, August 2010.
2. Albert Bifet, Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Indre Zliobaite. *CD-MOA: Change detection framework for massive online analysis.*, pages 92–103. Berlin: Springer, 2013.
3. Isvani Inocencio Frias Blanco, Jose del Campo-Avila, Gonzalo Ramos-Jimenez, Rafael Morales Bueno, Agustin Alejandro Ortiz Diaz, and Yaile Caballero Mota. Online and non-parametric drift detection methods based on hoeffding's bounds. *IEEE Trans. Knowl. Data Eng.*, 27(3):810–823, 2015.
4. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* Wiley, New York, 2. edition, 2001.
5. J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, in press, 2013.
6. Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In *In SBIA Brazilian Symposium on Artificial Intelligence*, pages 286–295. Springer Verlag, 2004.
7. Joao Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):44:1–44:37, March 2014.
8. Russell Greiner, Adam J. Grove, and Dan Roth. Learning cost-sensitive active classifiers. *Artif. Intell.*, 139(2):137–174, August 2002.
9. Frederik Gustafsson. *Adaptive Filtering and Change Detection.* Wiley, October 2000.
10. Maayan Harel, Shie Mannor, Ran El-yaniv, and Koby Crammer. Concept drift detection through resampling. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1009–1017. JMLR Workshop and Conference Proceedings, 2014.
11. Konrad Jackowski, Bartosz Krawczyk, and Michal Wozniak. Adass+ the hybrid training method of a classifier based on a feature space partitioning. *International Journal of Neural Systems,*, under review, 2013.
12. Bartosz Krawczyk. One-class classifier ensemble pruning and weighting with firefly algorithm. *Neurocomputing*, 150:490–500, 2015.
13. L.I. Kuncheva. Clustering-and-selection model for classifier combination. In *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on*, volume 1, pages 185 –188 vol.1, 2000.
14. Ludmila I. Kuncheva. Classifier ensembles for changing environments. In Fabio Roli, Josef Kittler, and Terry Windeatt, editors, *Multiple Classifier Systems, 5th International Workshop, MCS 2004, Cagliari, Italy, June 9-11, 2004, Proceedings*, volume 3077 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2004.
15. Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms.* Wiley-Interscience, 2004.

16. E. Lughofer and P.P. Angelov. Handling drifts and shifts in on-line data streams with evolving fuzzy systems. *Appl. Soft Comput.*, 11(2):2057–2068, 2011.
17. Zhenzheng Ouyang, Yuhai Gao, Zipeng Zhao, and Tao Wang. Study on the classification of data streams with concept drift. In *FSKD*, pages 1673–1677. IEEE, 2011.
18. E. S. Page. Continuous Inspection Schemes. *Biometrika*, 41(1/2):100–115, 1954.
19. Sarunas Raudys. *Statistical and Neural Classifiers: An Integrated Approach to Design*. Springer Publishing Company, Incorporated, 2014.
20. Jeffrey C. Schlimmer and Richard H. Granger, Jr. Incremental learning from noisy data. *Mach. Learn.*, 1(3):317–354, March 1986.
21. Raquel Sebastiao and Joao Gama. A study on change detection methods. *Progress in Artificial Intelligence, 14th Portuguese Conference on Artificial Intelligence, EPIA*, pages 12–15, 2009.
22. Piotr Sobolewski and Michal Wozniak. Concept drift detection and model selection with simulated recurrence and ensembles of statistical detectors. *Journal of Universal Computer Science*, 19(4):462–483, feb 2013.
23. Gerhard Widmer and Miroslav Kubat. Effective learning in dynamic environments by explicit context tracking. In PavelB. Brazdil, editor, *Machine Learning: ECML-93*, volume 667 of *Lecture Notes in Computer Science*, pages 227–243. Springer Berlin Heidelberg, 1993.
24. Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Mach. Learn.*, 23(1):69–101, April 1996.
25. Michal Wozniak. A hybrid decision tree training method using data streams. *Knowl. Inf. Syst.*, 29(2):335–347, 2011.
26. Michal Wozniak, Manuel Grana, and Emilio Corchado. A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3 – 17, 2014. Special Issue on Information Fusion in Hybrid Intelligent Fusion Systems.