Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 184 (2021) 314–321

Procedia Computer Science

www.elsevier.com/locate/procedia

The 12th International Conference on Ambient Systems, Networks and Technologies (ANT)
March 23 - 26, 2021, Warsaw, Poland

KAPPA as Drift Detector in Data Stream Mining

Osama A. Mahdi[a,*], Eric Pardede[a], Nawfal Ali[b]

[a] *La Trobe University, Melbourne, Vic 3086, Australia*
[b] *Monash University, Clayton, Vic 3800, Australia*

Abstract

Concept Drift is considered a challenging problem that appears in data streaming. The classifier's error rate and the ensemble are used in most of the previous works to manage classification accuracy as a criterion for judging whether concept drift is happening or not. KAPPA is an effective way to measure the level of agreement, and it may be suitable to detect concept drift in a reliable, fast, and computationally efficient way. In this paper, we propose a new concept drift detector, called KAPPA, which aims at reacting to detect concept drift in a reliable, fast, and computationally efficient way. Contrary the disagreement measure that we have already considered in our preliminary work (DMDDM), KAPPA would measure the level of agreement when different classifiers access data items is suitable to detect concept drifts. The performance of KAPPA has been experimentally compared with DMDDM on synthetic dataset streams, considering different performance measures, e.g., delay detection, true positives and the mean accuracy.

Keywords: Concept Drift; Data Stream Mining; Non-stationary Environments; KAPPA

1. Introduction

There are three major obstacles to learning from data streams [1, 2, 3, 4, 5, 6]: *variability*, *size*, and *speed*. The size and speed of data force algorithms to process them despite a restricted amount of memory and time, while examining each incoming instance only once [7, 8, 9]. Variability, on another hand, includes learning in environments that are dynamic with changing patterns. The most commonly studied explanation of variability in data streams is *concept drift*, i.e., changes in definitions and distributions of learned concepts over time [10]. Such unforeseen changes are mirrored in new learning instances which diminishes the accuracy of the algorithms trained from previous instances. For instance, take the example of examining a stream of microblog content regarding a movie in production. Upon changing the actor accountable for the key role, the stream of views regarding the film can rapidly be unfavorable. This problem is usually viewed as being concept drift in the sentiment of numerous groups of people. An algorithm trained

* Corresponding author. Tel.: +61-421-457-475 ; fax: +0-000-000-0000.
E-mail address: o.mahdi@latrobe.edu.au

1877-0509 © 2021 The Authors. Published by Elsevier B.V.

10.1016/j.procs.2021.03.040

on all the available content will recommend an overly optimistic regular opinion about the film [11, 12]. Thus, the data mining strategies which cope with concept drift are designed to execute forgetting, adaptation, or drift detection mechanisms to be able to adjust to changing environments.

Real-life examples of concept drift include evolving customer preferences, monitoring systems, weather predictions and financial fraud detection. Recently, the field of concept drift has attracted much research attention, where mining data streams that deal with concept drift should be capable of adapting to non-stationary environments. According to [2], four requirements need to be met by a model that operates in a non-stationary environment. Any predictive models must (*i*) detect a concept drift in a short time, (*ii*) differentiate noise from drift and be adaptive to changes but robust to noise, (*iii*) process in less time than incoming data arrival and, (*iv*) use no more than a constant amount of memory. A good data stream model/classifier should be able to combine these four requirements.

Different to the disagreement measure that is used in DMDDM [15], measuring the level of agreement when different classifiers access data items is suitable to detect concept drifts. The main goal of this paper is to answer the following research question:

- **Research Question:** *Contrary to the disagreement measure that was used in DMDDM, is measuring the level of agreement using KAPPA suitable to detect concept drift when different classifiers access data items?* The task is to signal concept drift in less time and with less memory consumption, keeping the accuracy of the data stream models constant.
- **Research Objective:** To do this, we empirically compare our proposed drift detector with our previously proposed drift detector DMDDM using synthetic data streams, considering different performance measures, e.g., delay detection, true positives and the mean accuracy.

We introduce KAPPA as the drift detector in data streams to address the aforesaid research question in the following sections.

The remainder of this paper is as follows. Section 2 presents the basic concepts and notation on concept drift. Section 3 describes the details of KAPPA. The analysis and the results of the synthetic and real datasets are presented in Section 4, followed by the conclusion and future work in Section 5.

2. Basic concepts and Notation

According to Bayesian decision theory [16], a classification model can be described by the prior probabilities of classes $p(y)$ and class conditional probabilities $p(x|y)$ for all classes $y \in \{K_1, \ldots, K_c\}$, where c is the number of predefined classes. Therefore, the classification decision for instance X at equal costs of mistake is made based on maximal a posteriori probability, which for class y can be represented as $p(y|X) = p(y)p(X|y)/p(X)$ where $p(X) = \sum_{y=1}^{c} p(y)p(X|y)$. In non-stationary environments, a data stream is reflected by changes in these probability distributions in an event called concept drift. Concept drift means that the concept about which data is being collected may shift from time to time after some minimal stable period [10][17]. Formally, concept drift between time point t^0 and time point t^1 can be defined as follows [2].

Definition 1. For a given data stream S, concept drift may occur between two points in time, t and $t + \Delta$, iff $\exists\, x$: $p^t(x, y) \neq p^{t+\Delta}(x, y)$ where p^t refers to the joint distribution at time t between the set of input attributes and the class label.

By considering this, any changes in incoming data can be characterized by changes the in components of the Bayesian decision theory [18, 19, 20]

- Prior probabilities $p(y)$ are prone to changes.
- Probabilities $p(X|y)$ of class conditional are also prone to changes.
- Consequently, posterior probabilities $p(y|X)$ may/may not change.

In accordance with these changes, as shown in **Fig.1**, two main kind of drifts are *virtual* and *real drift* [2, 21, 22]. Virtual drift is recognized by any change in the $p(x)$ / class $p(y)$ distributions which do not influence $p(y|X)$. Real drift is drift that is recognized by any changes in $p(y|X)$, noting that such changes may happen with/without changes in $p(x)$.

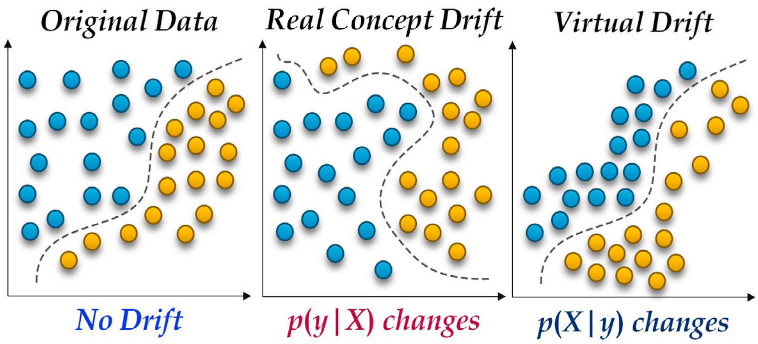

Fig. 1: Two types of drift [5]: where instances are represented by circles and different classes are represented by different colors.

To simplify the difference between real and virtual drift, let us take the example of the classification problem presented in **Table 1**. The task in the example is to determine whether a given flight will be delayed or not. If an airline company changes the flight time, but this does not result in a delay, this is regarded as virtual drift. Similarly, if due to a crisis, a company changes the frequency of certain flights but again, the flights leave without any delay, this is also regarded as virtual drift. However, if some flights are regularly delayed even though they used to be on time, real drift is occurring. The difference between real and virtual drift is illustrated in **Fig.1**, where the plot shows that only real concept drifts change the class boundary making any previously created model obsolete. The illustrated real drift occurs without any change in the attribute space, however, in practice, changes in prior probabilities may appear in combination with real drift.

Table 1: Sample from the airlines dataset/ where Airline=A, Flight=F, Day of week= Dow, Time=T, Length=L, Delayed=D.

A	F	From	To	DOW	T	L	D?
CO	269	SFO	IAH	Wed	15	205	yes
US	1558	PHX	CLT	Wed	15	222	yes
AA	2400	LAX	DFW	Wed	20	165	yes
AA	2466	SFO	DFW	Wed	20	195	yes
AS	108	ANC	SEA	Wed	30	202	no
CO	1094	LAX	IAH	Wed	30	181	yes

As we are mostly interested in the effect of concept drift on classification, we focus on methods that use true class labels to detect drift. We therefore concentrate on real drifts regardless as to whether they are visible from the input data distribution $p(x)$. In addition, researchers differentiate how these changes happen, as shown in **Fig.2**

A sudden/abrupt drift happens when suddenly the source distribution in S^t at a moment in time t is substituted by another distribution S^{t+1}. Once the new distribution has been used to train a generated classifier, a sudden drift would reduce the classification abilities of a classifier, whereas gradual drift is connected with a slower rate of change and it refers to a transition stage where examples of two different distributions P^j and P^{j+1} are mixed. With the passage of time, the likelihood of monitoring P^j examples decreases whereas the likelihood of monitoring P^{j+1} examples increases. Another kind of drift refers to recurrent concepts, i.e., after a period of time, previous concepts may reappear. The most recent approaches to address the three types of drifts are: abrupt drift [23] gradual drift [24] [25] and recurring drift [26, 27]. We emphasize that the proposed concept drift detection method has been designed mainly for sudden/abrupt drift.

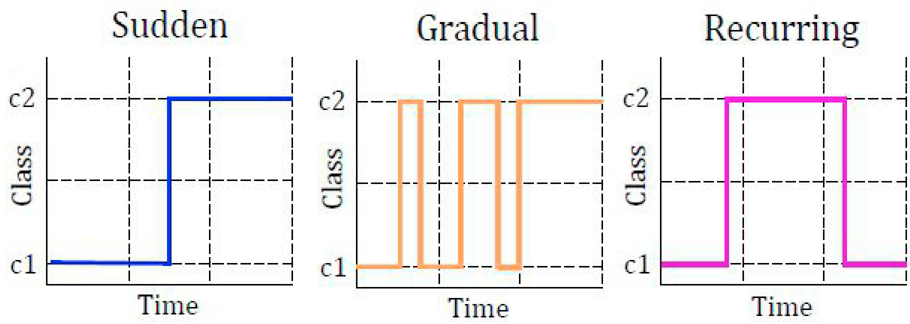

Fig. 2: Concept Drift Patterns [5]

3. Inter-rater Agreement KAPPA, in Data Streams

Learning from data streams in the presence of concept drift is one of the biggest challenges in contemporary machine learning. Since data class distributions may change through the progress of the stream, KAPPA provides better insight than the other metrics to detect concept drift in data class distribution. First, as KAPPA is a strict measure that quickly drops in case of incorrect predictions, this makes it much more useful than using accuracy/ error-rate which only introduces small changes. Second, the main reason concept drift occurs is due to changes/drifts in data class distribution as the stream progresses. KAPPA is capable of capturing the competence of the components reflecting the possibly varying data class distribution with time [13, 14] In addition, KAPPA is a statistic that is commonly used for handling the problem of imbalanced classification [28, 29, 30]. It evaluates the competence of a classifier by measuring the inter-rater agreement between successful predictions and the statistical distribution of data classes, correcting agreements that occur by mere statistical chance [31]. Formally, a statistic developed as a measure of inter-rater reliability, called k, can be used when different raters (here classifiers) assess data items to measure the level of agreement while correcting for chance [13]. For c class labels, k is defined on the C x C coincidence matrix M of the two classifiers. The entry of M is the proportion of the data set, which h_u labels as ω_k and h_v labels as ω_s. The agreement between h_u and h_v is given by

$$k_{i,j} = \frac{\sum_k m_{kk} - ABC}{1 - ABC} \tag{1}$$

where $\sum_{t=1} m_{kk}$ is the observed agreement between the classifiers and "ABC" is "agreement-by-chance"

$$ABC = \sum_k (\sum_s m_{k,s})(\sum_s m_{s,k}) \tag{2}$$

Low values of k signify higher disagreement and hence higher diversity. If calculated on the $2X2$ joined oracle output space using probabilities,

$$k_{i,j} = \frac{2(ac - bd)}{(a+b)(c+d) + (a+c)(b+d)} \tag{3}$$

For the purpose of this work, we divide Equation 3 by 2 as we are using a pair of classifier.

$$k_{i,j} = (\frac{2(ac - bd)}{(a+b)(c+d) + (a+c)(b+d)})/2 \tag{4}$$

The KAPPA approach is presented in Algorithm 1. First, the algorithm processes each example from the data stream and obtains the predictions for a pair of classifiers on each example line (lines 1-15). As the main idea behind these 15 steps, KAPPA tries to find all the possibilities which could exist using a pair of classifiers, as shown in Table 2. Then, the algorithm builds the oracle output table, as shown in Table 2. From Table 2, it can be seen that we need to find all cases (a, b, c and d). Once we observe all the cases, we count the number of observations on which the classifier is correct and incorrect for each case. Line (16) in our updated equation of KAPPA finds the agreement between the pair of classifiers. Then, the fading factor approach is applied from lines 17 to 19 (adopted from DMDDM [15]). In lines 17 and 18, the fading sum and fading increment are calculated, respectively. With the fading sum and fading increment, we use the value of KAPPA from line 16 as the observed value instead of the error estimates that were used in the original PH test. In line 19, the fading average is calculated. To monitor the diversity of a pair of classifiers, from lines 20 to 28, the PH test considers a variable m_T, which measures the accumulated difference between the observed value of diversity and their mean up to the current moment. After each monitoring, there is a step for checking and signaling for a drift, if the value of variation between the current m_T and the smallest value up to this moment M_T is larger than a predefined threshold. If there is a drift, evaluates the detected drift in terms of delay detection, true detection, false alarm and false negative. Finally, whether there is a drift or not, line 29 is used to incrementally train the model and keep it up-to-date.

Algorithm 1: Pseudocode of the KAPPA as Drift Detector in Data Stream

Require: S: data stream of examples (labelled),
Forgetting factor α : 0 <<α <-1
Admissible change: $\delta = 0.1$,
Drift threshold: $\lambda = 100$
M_T : 1.0D
Result: Drift ∈ {TRUE, FALSE}

1 **for** *each example* $x^t \in S$ **do**
2     $C_v prediction$ = get prediction using x^t;
3     $C_u prediction$ = get prediction using x^t;
4   **if** $C_v prediction$ *= 0.0 and* $C_u prediction$*= 1.0* **then**
5   b++;
6   **end**
7   **if** $C_v prediction$ *= 1.0 and* $C_u prediction$*= 0.0* **then**
8   c++;
9   **end**
10   **if** $C_v prediction$ *= 0.0 and* $C_u prediction$*= 0.0* **then**
11   a++;
12   **end**
13   **if** $C_v prediction$ *= 1.0 and* $C_u prediction$*= 1.0* **then**
14   d++;
15   **end**
16     $k_{i,j} = (2(ac - bd)/(a + b)(c + d) + (a + c)(b + d))/2$;
17     $S_{u:v,\alpha}(t) = k_{i,j} + \alpha \times S_{u:v,\alpha}(t - 1)$;
18     $N_\alpha(t) = 1 + \alpha \times N_\alpha(t - 1)$;
19     $M_\alpha(t) = \frac{S_{u:v,\alpha}(t)}{N_\alpha(t)}$;
20     $SumDiversity = SumDiversity + M_\alpha(t)$;
21     $m_T = (m_T + M_\alpha(t) - (SumDiversity/instancesSeen) - \delta)$;
22     $M_T = Min(M_T, m_T)$;
23     $PH_t est = m_T - M_T$;
24   **if** $PH_t est > \lambda$ **then**
25   Return TRUE
26   **else**
27   Return FALSE
28   **end**
29   incrementally train C_v and C_u with x^t;
30 **end**

Table 2: The Correlation of a Pair of Classifiers (2×2)

$h_u = h_v$	$h_u correct(1)$	$h_u incorrect(0)$
$h_v correct(1)$	a	b
$h_v incorrect(0)$	c	d

4. Experimental Evaluation

This subsection presents the outcomes of the analyses and the experiments of the proposed drift detector using three synthetic datasets where concept drifts are inserted in different locations. Table 3 epitomizes the characteristics of each dataset used in this work [1].

Table 3: Characteristics of Each Dataset.

Dataset	No.Inst	No.Attrs	No.Cls	Noise	No.Drifts	Drift Type	Drift Points
SEA 10K	100K	3	2	10%	1	sudden	10K
SEA 20K	100K	3	2	10%	1	sudden	20K
SEA 50K	100K	3	2	10%	1	sudden	50K
Mixed	100K	4	2	10%	4	sudden	20K
Sine1	100K	2	2	10%	4	sudden	20K

Fig 3 and Fig 4 compares the outcomes of the experiments using delay detection, true positives and mean accuracy based on five well-known datasets, namely: Mixed, Sine1, SEA 10K, SEA 20K and SEA 50K. First, in relation to the experiments using the Mixed and Sine1 datasets, for the Mixed dataset, DMDDM was able to detect four drifts successfully and slightly faster than KAPPA, 4 and 3.9 in terms of true positive (TP) and 35.11 and 4 in terms of delay detection, respectively. In addition, similar to the Mixed dataset, the results for Sine1 show that DMDDM is still slightly better than KAPPA in terms of delay detection, true positive and mean accuracy. On the other hand, the last three datasets, SEA 10K, SEA 20K and SEA 50K, showed very promising results, with KAPPA achieving significant results compared to DMDDM. From Fig 4 (a-c) and in terms of delay detection, KAPPA was much faster than DMDDM, whereas in terms of detecting drift, both KAPPA and DMDDM were successfully able to detect a single drift in all datasets. Finally, in terms of mean accuracy, both drift detectors achieve almost identical results.

Fig. 3: Performance of KAPPA using Mixed dataset VS Sine1 dataset.

[1] The proposed algorithm was implemented in Java as part of the MOA framework. The experiments were conducted on a machine equipped with Intel Core i7 @ 3.4 GHz with 16 GB of RAM running Windows 10. For KAPPA, we use the Hoeffding tree (HT) and Perceptron (PER) as our incremental classifiers.

(a) (b) (c)

Fig. 4: Performance of KAPPA using SEA10K VS SEA20K VS SEA50K dataset

5. Conclusion

We presented KAPPA as a drift detector in data stream mining, which was designed to react to sudden concept drift. Since data class distributions may change through the progress of the stream, KAPPA provides better insight than other metrics to detect concept drift. First, this is because KAPPA is a strict measure which quickly drop in the event of incorrect predictions, making it much more useful than using accuracy/ error-rate that only introduces small changes. Second, the main reason for concept drift is due to changes/drifts in data class distribution as the stream progresses. Therefore, KAPPA is capable of capturing the competence of the components reflecting the possibly varying data class distribution with time. The final results confirm that KAPPA efficiently handles and detects concept drifts, where the results showed it has a comparable performance with DMDDM. As future work, we plan to apply improvements in the proposed algorithm in order to cover other types of drifts, as well as aiming to investigate the possibility of adapting the proposed algorithm to work in partially labeled data.

References

[1] Wares, Scott, John Isaacs, and Eyad Elyan. "Data stream mining: methods and challenges for handling concept drift." *SN Applied Sciences* **1**, no. 11 (2019): 1412.

[2] Gama, Joao, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. "A survey on concept drift adaptation." *ACM computing surveys (CSUR)* **46**, no. 4 (2014): 1-37.

[3] Krempl, Georg, Indre Žliobaite, Dariusz Brzeziński, Eyke Hüllermeier, Mark Last, Vincent Lemaire, Tino Noack et al. "Open challenges for data stream mining research." *ACM SIGKDD explorations newsletter* **16**, no. 1 (2014): 1-10.

[4] Iwashita, Adriana Sayuri, and João Paulo Papa. "An overview on concept drift learning." *Ieee Access* **7** (2018): 1532-1547.

[5] Mahdi, Osama A., Eric Pardede, Nawfal Ali, and Jinli Cao. "Fast Reaction to Sudden Concept Drift in the Absence of Class Labels." *Applied Sciences* **10**, no. 2 (2020): 606.

[6] Mahdi, Osama A., Eric Pardede, and Jinli Cao. "Combination of information entropy and ensemble classification for detecting concept drift in data stream." In Proceedings of the Australasian Computer Science Week Multiconference, pp. 1-5. 2018.

[7] Domingos, Pedro, and Geoff Hulten. "Mining high-speed data streams." In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, 2000, pp. 71-80. .

[8] Street, W. Nick, and YongSeog Kim. "A streaming ensemble algorithm (SEA) for large-scale classification." In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, 2001, pp. 377-382.

[9] Kuncheva, Ludmila I. "Classifier ensembles for changing environments." In International Workshop on Multiple Classifier Systems, Springer, Berlin, Heidelberg, 2004, pp. 1-15.

[10] Gama, Joao. Knowledge discovery from data streams. CRC Press, 2010.

[11] Koren, Yehuda. "Collaborative filtering with temporal dynamics." In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining,2009, pp. 447-456.

[12] Ding, Yi, and Xue Li. "Time weight collaborative filtering." In Proceedings of the 14th ACM international conference on Information and knowledge management, 2005,pp. 485-492. .

[13] Kuncheva, Ludmila I. Combining pattern classifiers: methods and algorithms. John Wiley and Sons, 2014.

[14] Cano, Alberto, and Bartosz Krawczyk. "Kappa Updated Ensemble for drifting data stream mining." *Machine Learning* **109**, no. 1 (2020): 175-218.

[15] Mahdi, Osama A., Eric Pardede, Nawfal Ali, and Jinli Cao. "Diversity measure as a new drift detection method in data streaming." *Knowledge-Based Systems* **32** (2020): 105227.

[16] Duda, Richard O., Peter E. Hart, and David G. Stork. "Pattern classification." *International Journal of Computational Intelligence and Applications* **1** (2001): 335-339.

[17] Dong, Fan, Jie Lu, Guangquan Zhang, and Kan Li. "Active fuzzy weighting ensemble for dealing with concept drift." *International Journal of Computational Intelligence Systems* (2018).

[18] Gao, Jing, Wei Fan, Jiawei Han, and Philip S. Yu. "A general framework for mining concept-drifting data streams with skewed distributions." In Proceedings of the 2007 siam international conference on data mining, Society for Industrial and Applied Mathematics, 2007, pp. 3-14.

[19] Kelly, Mark G., David J. Hand, and Niall M. Adams. "The impact of changing populations on classifier performance." In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 367-371. 1999.

[20] Žliobaitė, Indrė. "Learning under concept drift: an overview." arXiv preprint arXiv:1010.4784 (2010).

[21] Lu, Jie, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. "Learning under concept drift: A review." *IEEE Transactions on Knowledge and Data Engineering* **31**, no. 12 (2018): 2346-2363.

[22] Hoens, T. Ryan, Robi Polikar, and Nitesh V. Chawla. "Learning from streaming data with concept drift and imbalance: an overview." Progress in *Artificial Intelligence* **1**, no. 1 (2012): 89-101.

[23] Pesaranghader, Ali, Herna L. Viktor, and Eric Paquet. "McDiarmid drift detection methods for evolving data streams." In 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1-9. IEEE, 2018.

[24] Liu, Anjin, Guangquan Zhang, and Jie Lu. "Fuzzy time windowing for gradual concept drift adaptation." In 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1-6. IEEE, 2017.

[25] Alippi, Cesare, Wen Qi, and Manuel Roveri. "Learning in nonstationary environments: A hybrid approach." In International Conference on Artificial Intelligence and Soft Computing, pp. 703-714. Springer, Cham, 2017.

[26] Ahmadi, Zahra, and Stefan Kramer. "Modeling recurring concepts in data streams: a graph-based framework." *Knowledge and Information Systems* **55**, no. 1 (2018): 15-44.

[27] Dhaliwal, Parneeta, and M. P. S. Bhatia. "Effective handling of recurring concept drifts in data streams." *Indian J. Sci. Technol* **10**, no. 30 (2017): 1-6.

[28] Ferri, César, José Hernández-Orallo, and R. Modroiu. "An experimental comparison of performance measures for classification." *Pattern Recognition Letters* **30**, no. 1 (2009): 27-38.

[29] Jeni, László A., Jeffrey F. Cohn, and Fernando De La Torre. "Facing imbalanced data–recommendations for the use of performance metrics." In 2013 Humaine association conference on affective computing and intelligent interaction, pp. 245-251. IEEE, 2013.

[30] Brzezinski, Dariusz, Jerzy Stefanowski, Robert Susmaga, and Izabela Szczech. "Visual-based analysis of classification measures and their properties for class imbalanced problems." *Information Sciences* **462** (2018): 242-261.

[31] Cano, Alberto, Amelia Zafra, and Sebastián Ventura. "Weighted data gravitation classification for standard and imbalanced data." *IEEE transactions on cybernetics* **43**, no. 6 (2013): 1672-1687.