Neurocomputing 452 (2021) 309–316

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

The prior probability in the batch classification of imbalanced data streams

Paweł Ksieniewicz [a]

[a] Department of Systems and Computer Networks, Wrocław University of Science and Technology, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland

ARTICLE INFO

Article history:
Received 31 May 2019
Revised 27 October 2019
Accepted 26 November 2019
Available online 20 November 2020
Communicated by Chennai Guest Editor

Keywords:
Pattern recognition
Classification
Imbalanced data
Data streams
Concept drift

ABSTRACT

In the diversity of contemporary decision-making tasks, where the data is no longer static and changes over time, data stream processing has become an important issue in the field of pattern recognition. In addition, most of the real problems are not balanced, representing their classes in various improportions. Following paper proposes the *Prior Imbalance Compensation* method, modifying on-the-fly predictions made by the base classifier, aiming at mapping *prior* probability in the statistics of assigned classes. It is intended to be a less computationally complex competition for popular algorithms such as SMOTE, solving this problem by oversampling the training set. The proposed method has been tested using computer experiments on the example of a set of various data streams, leading to promising results, suggesting its usefulness in solving this type of problems.

1. Introduction and related works

Pattern recognition methods are a collection of fundamental tools for solving problems faced by artificial intelligence, employing the popular interest that has been growing in recent times. As a rule, we divide them into two groups [1]. The first of them – *supervised learning* – is responsible for the description of new, yet unknown cases, on the basis of a set of patterns already known by labels [2]. In the case of the second group – *unsupervised learning* – the whole analysis takes place on a set of objects without any prior description.

The most popular problem of supervised learning is the classification, which, unlike the regression estimating the continuous value, assigns new objects to the set of discrete classes. The algorithms for solving this task have been developing intensively since the beginning of this field and have already formed a large group of methods, ranging from simple solutions such as the *Naive Bayes Classifier* [3] or *k-Nearest Neighbors* [4], through *decision trees* [5] and *forests* [6], *Support Vector Machines*, up to *neural networks*, with particular emphasis on the most-recently-fashionable *deep convolutional networks* [7].

1.1. Imbalanced data

The assumption of the most of classification algorithms is the equal occurrence of each of the considered classes [8]. This becomes problematic in the case when *imbalance ratio*, called also *prior* probability, is disturbed and one of the problem classes occurs much frequently than the others [9]. Data of this type are called *imbalanced data*. Due to the fact that the dominant majority of real decision problems, i.e. medical diagnostics, SPAM or fraud detection, presents imbalanced data, where, what should be emphasized, a less numerous class is the key from the perspective of the problem. Therefore, it became necessary to develop appropriate methods for counteracting the tendency of classifiers to favor the majority class [10].

Most of the proposed solutions to the problem of imbalanced data may be assigned to one of three groups. The first, theoretically the simplest, are the mechanisms built directly in the process of classifier training, modifying their model to align the impact of all classes of the problem, for example by the use of appropriate loss function. The second and the most common approach is the appropriate preprocessing of training data to align the presence of problem class patterns in it. Above simple *random oversampling* and – *undersamplig* one should distinguish here the SMOTE [11] – algorithm generating synthetic samples, along with its numerous variants, and ADASYN [12], extending it to include the distribution of the majority class in the synthesis. The last, but not least, approach are hybrid methods [13], using group of diversified classifiers in the construction of the decision system [14] connected by the prior-sensitive decision rule [15].

https://doi.org/10.1016/j.neucom.2019.11.126
0925-2312/

1.2. Data streams and concept drift

Important in the context of real problems of classification is also the aspect of knowledge historicity. A classification that is completely correct at a given point in time may lose its validity in the future and eventually turn out to be wrong. Therefore, it is naïve to assume that once-trained model, used for a long time will induct an error – which once estimated – will not increase over time, and the classifier itself, will not be outdated.

In many problems, we do not deal with static data set, and in addition to attributes, objects are characterized by their location in time. Such cases are called data streams and we process them, in principle, in one of two ways. In the first of them – online processing – each incoming object is analyzed separately, one by one and in this mode it drives the updating of the classification model. However, it is a very computationally intensive approach, and so-called batch processing is used much more frequently. The principle of batch processing is that incoming patterns are accumulated in so-called chunks and processed not pattern by pattern, but group by group.

Among the solutions to the stream classification problem, the most popular are approaches that allow for partial model fitting, i.e. modifying the existing model with information extracted from upcoming data, like WINNOW [16] or VFDT [17], and the ensemble approach, especially popular in batch processing [18]. Employing the incremental learning methods requires implementation of forgetting mechanisms, either as built-in capabilities of algorithm [19] or as dataset weighting or windowing [20].

In such ensembles, successive members of the committee are built on the basis of subsequent chunks, making it possible to weigh the influence of the member decision on the final prediction according to their quality determined on the latest data, and to trim the committee in order to eliminate obsolete models [21].

The already mentioned aspect of knowledge historicity introduces an additional complication in the problem of data stream classification. Outdating of models with passing time is the result

Table 1
Average results of *Gaussian Naive Bayes* classifier depending on the processing parameters. Bolded are the **measurements significantly better** in the pair of the base method and modified by PIC.

MINORITY	BALANCED ACCURACY				F-SCORE			
CLASS	GRADUAL		SUDDEN		GRADUAL		SUDDEN	
PERCENTAGE	GNB	PIC	GNB	PIC	GNB	PIC	GNB	PIC
5%	0.627	**0.664**	0.694	**0.718**	**0.966**	0.959	**0.964**	0.961
10%	0.673	**0.707**	0.741	**0.759**	**0.943**	0.934	**0.944**	0.939
20%	0.722	**0.744**	0.779	**0.790**	**0.899**	0.890	**0.908**	0.903
50%	**0.766**	**0.766**	**0.805**	**0.804**	**0.764**	**0.758**	**0.802**	0.796

Table 2
Average results of *k-NN* classifier depending on the processing parameters. Bolded are the **measurements significantly better** in the pair of the base method and modified by AIC.

MINORITY	BALANCED ACCURACY				F-SCORE			
CLASS	GRADUAL		SUDDEN		GRADUAL		SUDDEN	
PERCENTAGE	k-NN	PIC	k-NN	PIC	k-NN	PIC	k-NN	PIC
5%	0.657	**0.763**	0.693	**0.787**	**0.979**	0.973	**0.981**	0.975
10%	0.756	**0.829**	0.787	**0.846**	**0.968**	0.963	**0.971**	0.966
20%	0.842	**0.874**	0.861	**0.886**	**0.952**	0.948	**0.956**	0.952
50%	**0.900**	**0.900**	**0.909**	0.908	**0.900**	0.898	**0.909**	0.905

Table 3
Average results of *Support Vector Classifier* depending on the processing parameters. Bolded are the **measurements significantly better** in the pair of the base method and modified by AIC.

MINORITY	BALANCED ACCURACY				F-SCORE			
CLASS	GRADUAL		SUDDEN		GRADUAL		SUDDEN	
PERCENTAGE	SVM	PIC	SVM	PIC	SVM	PIC	SVM	PIC
5%	0.624	**0.797**	0.670	**0.820**	**0.978**	0.977	**0.981**	0.979
10%	0.730	**0.844**	0.769	**0.862**	**0.968**	0.967	**0.972**	0.970
20%	0.827	**0.879**	0.851	**0.891**	**0.952**	0.950	**0.957**	0.954
50%	**0.900**	**0.899**	**0.908**	0.907	**0.900**	0.897	**0.908**	0.904

Table 4
Average results of *Random Forest Classifier* depending on the processing parameters. Bolded are the **measurements significantly better** in the pair of the base method and modified by AIC.

MINORITY	BALANCED ACCURACY				F-SCORE			
CLASS	GRADUAL		SSUDDEN		GRADUAL		SUDDEN	
PERCENTAGE	RFC	PIC	RFC	PIC	RFC	PIC	RFC	PIC
5%	0.643	**0.738**	0.696	**0.776**	**0.977**	0.971	**0.979**	0.974
10%	0.739	**0.805**	0.785	**0.834**	**0.965**	0.959	**0.968**	0.963
20%	0.828	**0.855**	0.856	**0.875**	**0.945**	0.940	**0.951**	0.946
50%	**0.884**	**0.883**	**0.899**	0.898	**0.883**	**0.881**	**0.898**	0.895

of the phenomenon called *concept drift* [22]. Among the concept drifts you may distinguish between *sudden drift*, where the change between class distributions occurs rapidly at precise point, as well as *incremental* or *gradual drift*, where the concepts of classes are changing smoothly [22,18]. Solutions to this problem are focused either on the drift detection, signaling the need to rebuild the model, or on the classifier ensemble. It is also important to mention the propositions how to react to detected drifts, like DWM [23], STAGGER [24], or GT2FC [25]. Appearances of *concept drift* in *data streams* have become a challenge for plethora of practical solutions, such as computer systems security [26,27], medical diagnosis [28] or fraud detection [29].

1.3. Contributions

The following work is intended to achieve the following goals:

- Proposal of a strategy for interpreting the support obtained on a batch of data by the probabilistic base classifier in a manner that takes into account the prior probability.

Table 5
Summary of the classification quality for each of the eight analyzed classifiers. Bolded are the **measurements significantly better** in the competition.

MINORITY CLASS PERCENTAGE	Base method GNB	k-NN	SVM	RFC	PIC GNB	k-NN	SVM	RFC
GRADUAL DRIFT								
5%	0.627	0.657	0.624	0.643	0.664	0.763	**0.797**	0.738
10%	0.673	0.756	0.730	0.739	0.707	0.829	**0.844**	0.805
20%	0.722	0.842	0.827	0.828	0.744	0.874	**0.879**	0.855
50%	0.766	**0.900**	**0.900**	0.884	0.766	**0.900**	**0.899**	0.883
SUDDEN DRIFT								
5%	0.694	0.693	0.670	0.696	0.718	0.787	**0.820**	0.776
10%	0.741	0.787	0.769	0.785	0.759	0.846	**0.862**	0.834
20%	0.779	0.861	0.851	0.856	0.790	0.886	**0.891**	0.875
50%	0.805	**0.909**	**0.908**	0.899	0.804	0.908	0.907	0.898

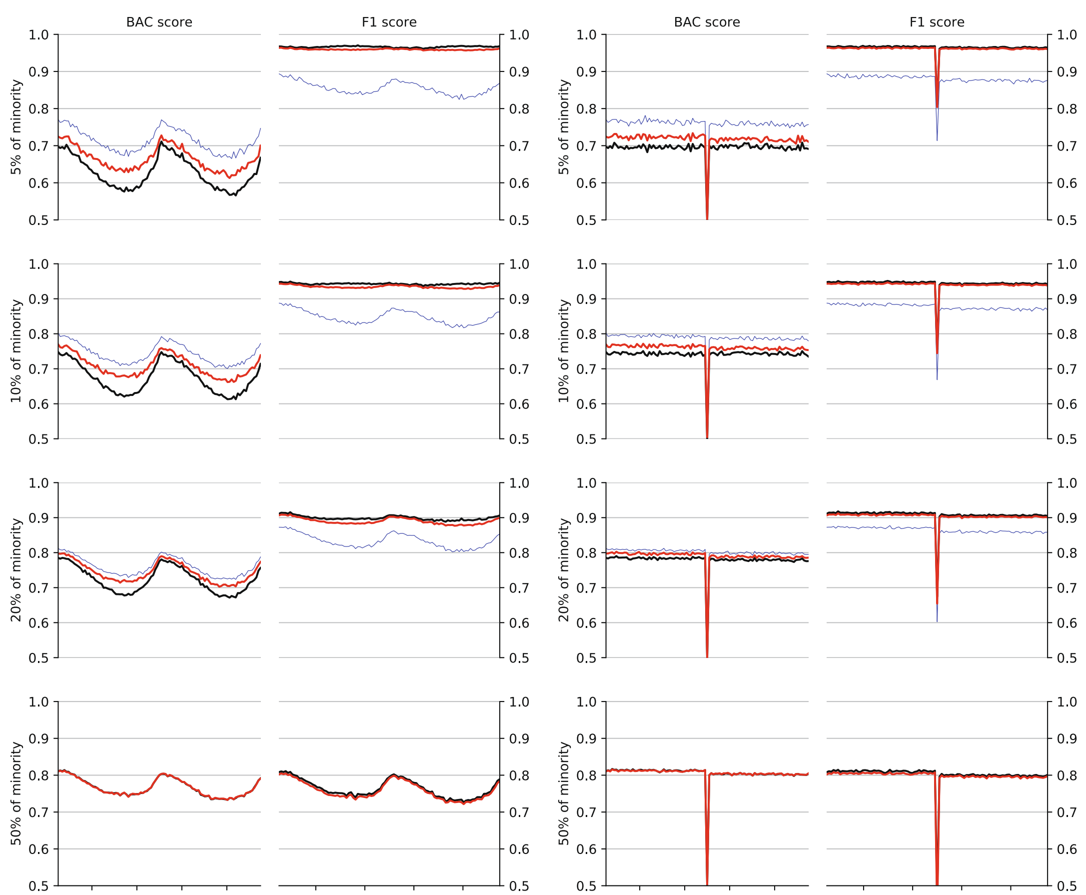


Fig. 1. Comparison of *balanced accuracy* and *F1 score* for bare *Gaussian Naive Bayes classifier* (black) and its proposed PIC version (red). *Gradual drift* on the left and *sudden drift* on the right side. Rows indicating different imbalance ratios (highly imbalanced stream on top, balanced data on bottom). For comparative purposes a blue line shows results with use of SMOTE oversampling in the same processing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- Experimental evaluation of the proposed approach on the example of a collection of benchmark data streams with various level of class imbalance, compared to the method without modification.

2. Employing prior probability of training set in batch classification

The method proposed in the following paper may be used together with any approach to batch processing of stream data, so for the clarity of the description and objectivity of experiments, it is presented with the simplest system of this type. We assume in it, in accordance with the principle of *test-and-train* evaluation, that for each incoming data chunk a single new model is built and it is used to predict the next batch of samples.

It is important to point out that the prerequisite for applying the described approach is to use the probabilistic model as a base classifier. This means that the answer of the trained algorithm to a new object is to be not only a direct assignment to one of the problem classes, but a *support vector* (*posterior probability* for the pattern) determining the chance of belonging to each class separately. Examples of such algorithms for data quantitative features may be *Gaussian Naive Bayes*, the *k-NN classifier* (with particular emphasis on its variant assigning distance-based weight to the set of nearest neighbors) or *Support Vector Machines* in its probabilistic interpretation.

2.1. Training

The first step of processing is to determine the local class distribution in the training set, which we carry out with each learning process, and thus in the considered case, with each subsequent chunk. The *prior* is the averaged value of this distribution on the current course of data stream. In contrast to the most of popular approaches, we do not modify the training set in any way (which is the case with all oversampling methods) but only extract simple statistical information from it.

2.2. Prediction

The second element of processing is the minimal modification of the prediction method based on the obtained supports. In the classic approach, the prediction is made independently for each pattern from the chunk, and the object is assigned to the class whose probability value was the largest. In the case of batch processing, however, we do not make prediction for a single sample, but for a set of patterns. This creates the potential to use information about the *prior* probability of the problem.

Sorting predicted objects, according to the support obtained for them, allows for assigning them to classes, using *posterior* probability, but keeping the class proportions consistent with *prior* probability. However, the proposed modification of predictions takes place only in cases where the *minority class* is underrepresented. In the case where the percentage of predictions in its favor is greater than according to stored *prior* probability, there is no artificial reduction. This is a delicate bias of the model in a favor of a more valuable class from the perspective of most classification problems.

2.3. The prior imbalance compensation

The proposed method, for the needs of the rest of this work, called *Prior Imbalance Compensation* (PIC), has a low computational overhead and was designed for data streams in which new objects appear with high frequency and in large numbers, or in cases where the imbalance ratio is very high (percentage of a minority class with less than 5% of objects), where in the absence of a sufficient number of minority examples it is not possible to generate synthetic patterns through the SMOTE algorithm or similar synthesizer. Overview of the proposed method is presented in Algorithm 1.

Algorithm 1: Prior Imbalance Compensation

Require:
data stream,
n – data chunk size,
training_procedure() – classifier training procedure,
classifier() – classification model,
$F_0()$ – support function of minority class used by *classifier*(),
$F_1()$ – support function of majority class used by *classifier*()

PREPARING INITIAL MODEL
chunk $\leftarrow$ initial chunk in data stream
a_priori $\leftarrow$ percentage of minority class samples in *chunk*
classifier $\leftarrow$ *training_procedure*(*chunk*)

STREAM PROCESSING LOOP
for *chunk* **in** data stream **do**
prediction $\leftarrow F_1() > F_0()$

PRIOR IMBALANCE COMPENSATION
support_order $\leftarrow$ *argsort*($F_0()$)
minority_treshold $\leftarrow$ *a_priori* $*\ n$
prediction[*support_order* $<$ *minority_treshold*] $\leftarrow 0$

ESTABLISHING A NEW MODEL
local_a_priori $\leftarrow$ percentage of minority class samples in *chunk*
a_priori $\leftarrow$ mean of all previous *local_a_priories*
classifier $\leftarrow$ *training_procedure*(*chunk*)
end for

3. Experiments set-up

A key issue in planning experiments on imbalanced data is the selection of appropriate evaluation measures. The use of classical accuracy in this case will give a completely non-quantifiable result, because for example, in the case of a blind classifier always returning the prediction towards the majority class, it will respond with its percentage in the test set, which will give – with imbalance ratio 1:20 – a very good, though hypocritical result of 95%.

From the standard metrics for imbalanced problems, two measures were selected for the experiment:

- F1-score, returning a ratio between the doubled sum and the product of precision and recall.
- Balanced accuracy score, which is the average recall for each class of problem.

An improvement in *balanced* accuracy in the absence of *F1-score* degeneration will be recognized as a promising result.

In the experiments, a standard *Test-and-train* approach to evaluation of data streams was used. It consists of, after training the initial model on the first chunk, on subsequent testing on the upcoming data chunk and transferring it to the next learning loop until the end of the data stream.

Four algorithms were selected as the base classifiers for processing:

- GNB – *Gaussian Naive Bayes*,
- **k-**NN – *k-Nearest Neighbors* with $k = 5$ in version with weights of neighbors according to their Minkowski distance from sampled point in decision space,
- SVM – *Support Vector Machines* in its probabilistic interpretation,
- RF – *Random Forest* with 20 estimators and *Gini* criterion.

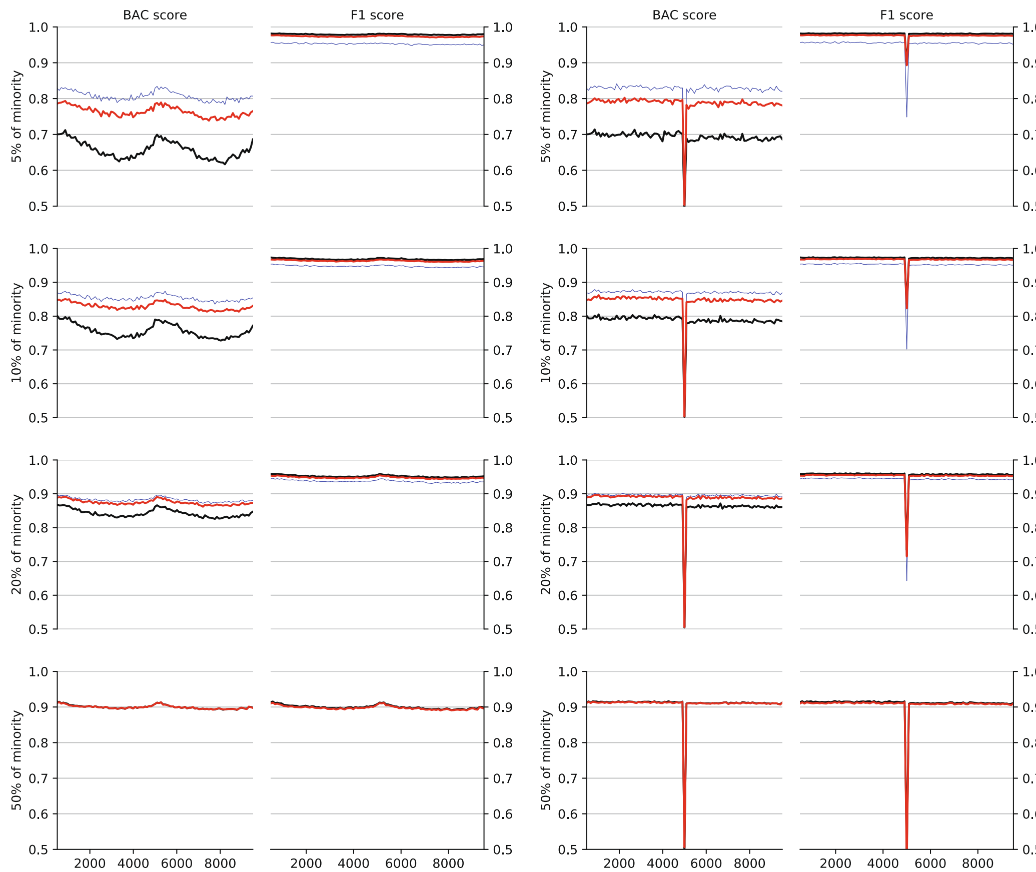


Fig. 2. Comparison of *balanced accuracy* and *F1 score* for bare *k-NN classifier* (black) and its proposed PIC version (red). *Gradual drift* on the left and *sudden drift* on the right side. Rows indicating different imbalance ratios (highly imbalanced stream on top, balanced data on bottom). For comparative purposes a blue line shows results with use of SMOTE oversampling in the same processing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

To ensure the most reliable estimation of the solutions quality, one hundred data streams consisting of 50,000 objects broken into 100 chunks of 500 objects (5 million in total) were generated, each containing two concepts established according to the rules of generating synthetic classification problems used to create the Madelon set [30]. Concept changes are made in two ways, using *sudden drift* and *incremental drift*. For each of these combinations, four variants of streams were prepared, with different *imbalance ratios* (successively 5, 10, 20 and 50% share of the minority class).

In total, it gives 800 data streams used in the evaluation, and the presented results, for the stabilization of the sampling and measurability, represent the average of each combination of the imbalanced ratio and the type of drift from the hundred runs.

The statistical dependence test results were based on the *Wilcoxon test*, and the whole implementation of the experiments was carried out in *Python 3*, based on the *scikit-learn* [31] library. The whole code necessary to repeat the experiments contained in this work is in the public *git* repository[1].

4. Experimental evaluation

4.1. Goals

The main goal of conducted experimental evaluation was to verify the impact of *Prior Imbalance Correction* on discriminative power of four probabilistic classifiers on collection data streams with various types of concept drifts and various scales of data imbalance.

4.2. Results

Illustrations 1–4 illustrate the *balanced accuracy* and the *F1 score* for all previously described combinations of data streams for the selected four base classifiers. They are supplemented, for comparative purposes, with quality readings marked with a blue line for use of oversampling in the same processing with the SMOTE algorithm. All the charts, due to the considered binary problem, were scaled from the level of the *random classifier* (50%) to the end of the scale (100%).

Clean concepts in *gradual drifts* locate near the beginning of the graph, its end and the exact center, and the greatest mixing during the *concept drift* takes place at points 3000 and 8000. The *sudden drifts* occur exactly at 5000 points.

Analogously, Tables 1–4 present a summary of the results averaged for each stream type, both for the base classifiers and their versions modified by the PIC. Table 5 provides a summary of the classification quality for each of the eight analyzed classifiers (four base classifiers with and without PIC modification) in each of the analyzed groups of data streams, along with an analysis of the statistical dependence of the achieved results.

4.3. Observations

Analyzing the results achieved for GNB as the base classifier (Table and Fig. 1), it may be observed that for the imbalanced

[1] https://github.com/w4k2/apriori-stream

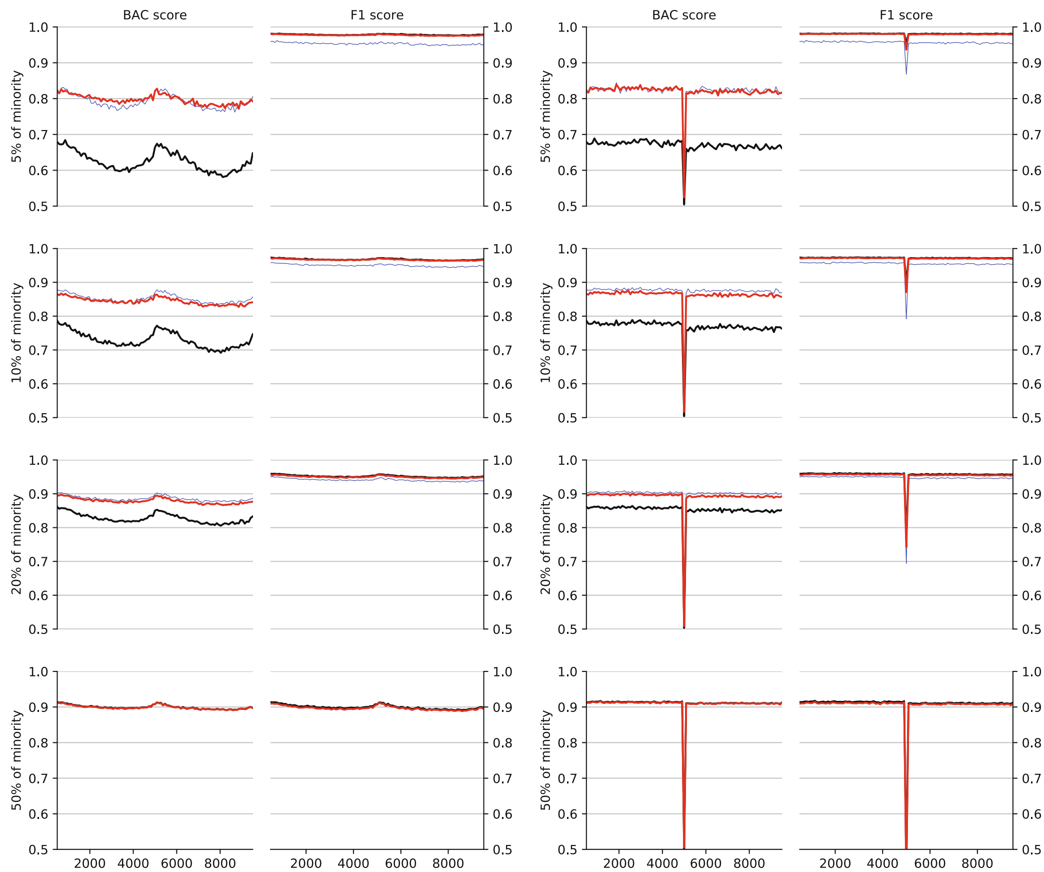


Fig. 3. Comparison of *balanced accuracy* and *F1 score* for bare *Support Vector Classifier* (black) and its proposed PIC version (red). *Gradual drift* on the left and *sudden drift* on the right side. Rows indicating different imbalance ratios (highly imbalanced stream on top, balanced data on bottom). For comparative purposes a blue line shows results with use of SMOTE oversampling in the same processing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

streams (regardless of the *imbalance ratio*), both gradual and sudden, the PIC method is characterized by a few percent improvement in the *balanced accuracy*, especially noticeable (and growing to about ten percent) in the moments of the strongest mixing of patterns during gradual drift. The *F1-score* measure shows only a slight degradation at the level of a few promilles.

It is interesting to compare the quality achieved by PIC (red) and SMOTE (blue line). Oversampling allows for even stronger increase of *balanced accuracy*, but it strongly degenerates *F1-score*.

A similar situation occurs for the use of the *k*-NN algorithm (Table and Fig. 2), where the improvement of PIC against the base method is even more noticeable for *balanced accuracy*, with equally low *F1-score* degeneration. The relation between PIC and SMOTE remains unchanged.

Of particular interest is the behavior of algorithms interpreted with PIC using the SVM algorithm (Table and Fig. 3), which, according to the observations contained in Table 5, achieves the statistically significant highest results from all competitors in the case of imbalanced streams, regardless of the scale of unbalancing. Modification of the PIC in this case gives results very similar to SMOTE, without the need for oversampling. In the case of a very strong imbalance ratio (5%), the results achieved, both according to the *balanced accuracy* and *F1-score* metrics, PIC turns out to be better than SMOTE.

A reasonable explanation for the above observation may be the principle of SMOTE algorithm operation, which does not seem to be an ideal solution in the case of *linear classifiers*, or the ones using *radial basis function* kernel. Employing it in processing moves the decision boundary of the constructed model in the direction taking into account the class imbalance, but by the relatively high impact of synthetic patterns, it always distorts its shape with a greater or lesser negative effect on the quality of classification. Approach, as in PIC, just scaling the prediction threshold at a given level, also makes the desired shift of the decision boundary, but does not affect its shape in any manner, which gives a solution more fitted to real training data and closer to the optimal one. Employment of RFC as a base algorithm (Table and Fig. 4), in the relation of processing methods, gives results similar to those achieved by SVM.

5. Conclusions

This work proposed the *Prior Imbalance Compensation* (PIC) method for use in batch processing of imbalanced data streams. It consists in determining the missing predictions of a minority class in the case of under-representation in the prediction of a chunk.

The algorithm achieves promising results, showing not only the advantage over the used base methods, but also the over more computational-costly method of SMOTE, without the need to generate synthetic samples that carry the risk of introducing incorrect patterns into the training set.

A possible disadvantage of the proposed method is the naive assumption of *prior* constancy in the entire flow of the data stream. Due to the promising results of the presented research, its variant resistant to variable *imbalance ratio* and the version adapted to online classification will be implemented in the near future.

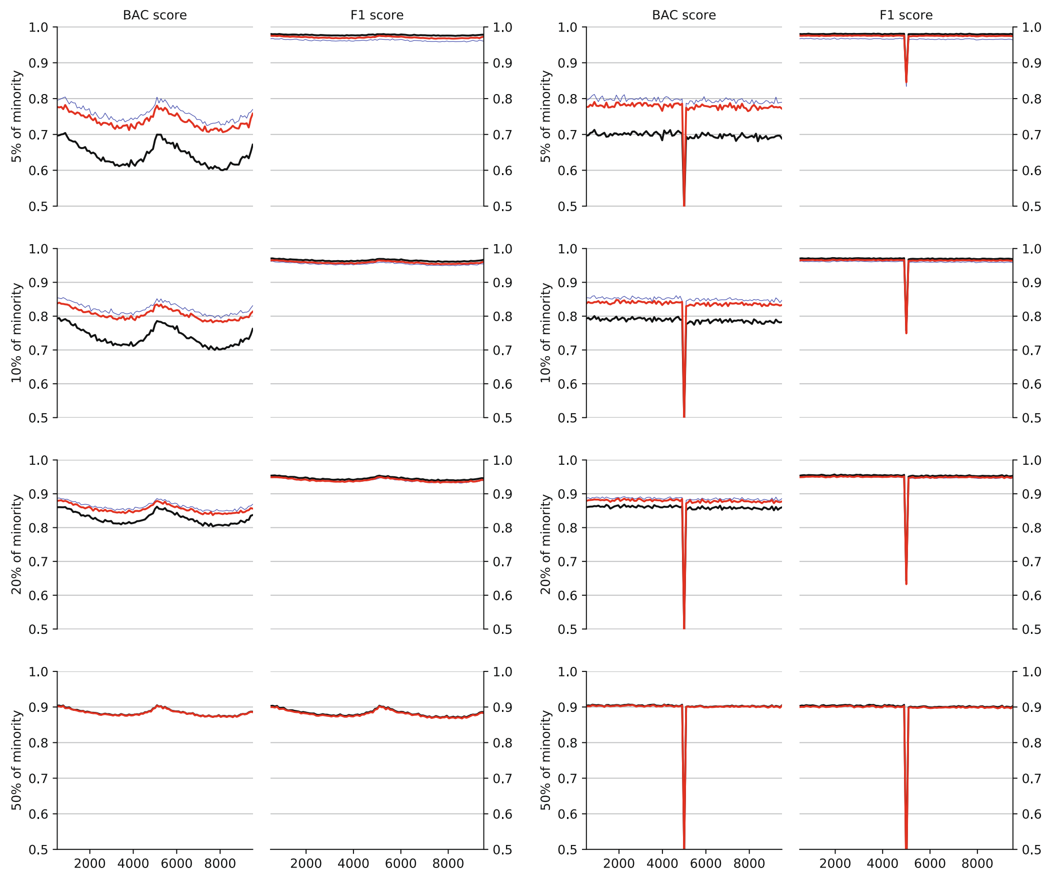


Fig. 4. Comparison of *balanced accuracy* and *F1 score* for bare *Random Forest Classifier* (black) and its proposed PIC version (red). *Gradual drift* on the left and *sudden drift* on the right side. Rows indicating different imbalance ratios (highly imbalanced stream on top, balanced data on bottom). For comparative purposes a blue line shows results with use of SMOTE oversampling in the same processing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

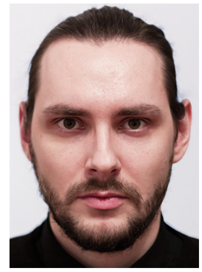
Acknowledgement

This work is supported by the Polish National Science Center under the Grant No. 2017/27/B/ST6/01325 as well the statutory funds of the Department of Systems and Computer Networks, Faculty of Electronics, Wrocław University of Science and Technology.

References

[1] T. Mitchell, B. Buchanan, G. DeJong, T. Dietterich, P. Rosenbloom, A. Waibel, Machine learning, Annu. Rev. Comput. Sci. 4 (1990) 417–433.
[2] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley, New York, 2001.
[3] P. Cheeseman, M. Self, J. Kelly, J. Stutz, W. Taylor, D. Freeman, AutoClass: a Bayesian classification system, in: Machine Learning: Proceedings of the Fifth International Workshop, Morgan Kaufmann, 1988..
[4] D. Wettschereck, T.G. Dietterich, An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms, Mach. Learn. 19 (1995) 5–27.
[5] D. Jankowski, K. Jackowski, B. Cyganek, Learning decision trees from data streams with concept drift, Procedia Comput. Sci. 80 (2016) 1682–1691.
[6] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 832–844.
[7] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436.
[8] Y. Sun, A.K.C. Wong, M.S. Kamel, Classification of imbalanced data: a review, Int. J. Pattern Recogn. Artif. Intell. 23 (2009) 687–719.
[9] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (2009) 1263–1284.
[10] N.V. Chawla, N. Japkowicz, A. Kotcz, Special issue on learning from imbalanced data sets, ACM Sigkdd Explor. Newslett. 6 (2004) 1–6.
[11] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[12] H. He, Y. Bai, E.A. Garcia, S. Li, Adasyn: adaptive synthetic sampling approach for imbalanced learning, in: Proceedings of the International Joint Conference on Neural Networks, pp. 1322–1328..
[13] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: 2009 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009 – Proceedings, pp. 324–331..
[14] P. Ksieniewicz, Combining random subspace approach with smote oversampling for imbalanced data classification, in: Hybrid Artificial Intelligent Systems – 14th International Conference, HAIS 2019, Leon, Spain, September 4–6, 2019, Proceedings..
[15] P. Ksieniewicz, M. Woźniak, Imbalanced data classification based on feature selection techniques, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, pp. 296–303..
[16] N. Littlestone, Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm, Mach. Learn. 2 (1988) 285–318.
[17] P. Domingos, G. Hulten, Mining high-speed data streams, in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '00, ACM, New York, NY, USA, 2000, pp. 71–80..
[18] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Computing Surveys in Press, 2013..
[19] P. Ksieniewicz, M. Woźniak, B. Cyganek, A. Kasprzak, K. Walkowiak, Data stream classification using active learned neural networks, Neurocomputing (2019).
[20] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (2014) 44:1–44:37..
[21] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[22] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, Mach. Learn. 23 (1996) 69–101.
[23] J. Kolter, M. Maloof, Dynamic weighted majority: a new ensemble method for tracking concept drift, in: Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, pp. 123–130..

[24] J.C. Schlimmer, R.H. Granger Jr., Incremental learning from noisy data, Mach. Learn. 1 (1986) 317–354.
[25] A. Bouchachia, C. Vanaret, GT2FC: an online growing interval type-2 self-learning fuzzy classifier, IEEE Trans. Fuzzy Syst. 22 (2014) 999–1018.
[26] T. Lane, C.E. Brodley, Approaches to online learning and concept drift for user identification in computer security, in: R. Agrawal, P.E. Stolorz, G. Piatetsky-Shapiro (Eds.), Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, New York, USA, August 27–31, 1998, AAAI Press, 1998, pp. 259–263..
[27] J.R. Méndez, F. Fdez-Riverola, E.L. Iglesias, F. Díaz, J.M. Corchado, Tracking Concept Drift at Feature Selection Stage in SpamHunting: An Anti-spam Instance-Based Reasoning System, Springer, Berlin Heidelberg, Berlin, Heidelberg, pp. 504–518..
[28] A.A. Beyene, T. Welemariam, M. Persson, N. Lavesson, Improved concept drift handling in surgery prediction and other applications, Knowl. Inf. Syst. 44 (2015) 177–196.
[29] A.D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, G. Bontempi, Credit card fraud detection and concept-drift adaptation with delayed supervised information, in: IJCNN, IEEE, 2015, pp. 1–8.
[30] I. Guyon, Design of experiments of the nips 2003 variable selection benchmark, in: NIPS 2003 Workshop on Feature Extraction and Feature Selection..
[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.



Paweł Ksieniewicz is an assistant professor of computer science at the Department of Systems and Computer Networks, Wroclaw University of Science and Technology, Poland. He received M.Sc. and Ph.D. degrees in computer science from the Wroclaw University of Science and Technology in 2013 and 2017, respectively. His research focuses on pattern classification and machine learning methods and computer vision.