Prior Probability Estimation in Dynamically Imbalanced Data Streams

1st Joanna Komorniczak
Department of Systems and Computer Networks
Wroclaw Univ. of Science and Technology
Wrocław, Poland
e-mail: 241245@student.pwr.edu.pl
ORCID: 0000-0002-1393-3622

2nd Paweł Zyblewski
Department of Systems and Computer Networks
Wroclaw Univ. of Science and Technology
Wrocław, Poland
e-mail: pawel.zyblewski@pwr.edu.pl
ORCID: 0000-0002-4224-6709

3st Paweł Ksieniewicz
Department of Systems and Computer Networks
Wroclaw Univ. of Science and Technology
Wrocław, Poland
e-mail: pawel.ksieniewicz@pwr.edu.pl
ORCID: 0000-0001-9578-8395

2021 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-3900-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IJCNN52387.2021.9533795

***Abstract*—Despite the fact that real-life data streams may often be characterized by the dynamic changes in the *prior class probabilities*, there is a scarcity of articles trying to clearly describe and classify this problem as well as suggest new methods dedicated to resolving this issue. The following paper aims to fill this gap by proposing a novel data stream taxonomy defined in the context of *prior class probability* and by introducing the *Dynamic Statistical Concept Analysis* (DSCA) – *prior probability* estimation algorithm. The proposed method was evaluated using computer experiments carried out on 100 synthetically generated data streams with various class imbalance characteristics. The obtained results, supported by statistical analysis, confirmed the usefulness of the proposed solution, especially in the case of *discrete dynamically imbalanced data streams* (DDIS).**

***Index Terms*—data stream, imbalanced data, dynamically imbalanced data stream, pattern recognition**

I. Introduction

The modern world is often described by both fiction [1] and scientific authors as a never-ending, diverse data stream controlling our lives. The reality increasingly confirms this supposition in the era of the digitization of our everyday life – accelerated by the coronavirus pandemic – in which most of our interpersonal contacts, cultural works we receive, our financial transactions or even home lighting control are carried out with streaming information sent via computer networks [2].

The data stream processing is a problem widely discussed in the literature [3]. The two main difficulties recurring in the analyzes are (*a*) unusually large data volumes, which forces the authors of processing algorithms to adapt to the rule of single processing of each pattern coming from a data stream [4], [5], and (*b*) the constantly changing probabilistic characteristics of the data stream, most often interpreted as a *concept drift* phenomenon, involving changes in the *posterior* class distributions in the time domain [6].

The vast majority of works in the field of non-stationary data stream processing deal with problems of changes in the *posterior* probability [7], relatively rarely addressing the topic of imbalanced streams, and in particular, dynamically imbalanced streams, i.e. those characterized by changes in the *prior probability* [8]. Meanwhile, a large part of data streams – with network streams as a representative example – can be characterized primarily by a huge volume in a relatively short period, which means that the influence of concept drift is stretched over time, reduced and sometimes even negligible. In such cases, changes in the prior class distribution become much more important [9].

The above-mentioned data may therefore be defined as static in terms of the concept, with dynamic proportion of individual classes, depending on the point in time at which we make the snapshot. The very type of differences in *prior probability* can also vary. There exists possible cases of problems that are globally balanced (on the entire analyzed data stream course), but are characterized by a strong disproportion of the class count in local data portions (*batches*).

Information about the prior distribution of problem classes may have a real impact on the quality of classification models built in the stream environment. The usefulness of such knowledge in statistically significant improvement of the imbalanced data stream classification has been substantiated in [10]. However, the mentioned work deals with only one simple method of *prior probability* estimation, which, cannot be generalized to all types of imbalanced data streams. We would like to propose the following taxonomy of data streams in the context of *prior class probability* (presented also in Figure 1):

- **BS** (*balanced streams*)
  The global prior and each of the local priors has a **proportional** and **dependent** class distribution,

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

- **SIS** (*statically imbalanced streams*)
  The global prior and each of the local priors are characterized by a **disproportionate** but **constant and dependent** class distribution,
- **DIS** (*dynamically imbalanced streams*)
  Among which we can distinguish two subcategories:
  - **CDIS** (*continuous dynamically imbalanced streams*)
    The global prior **may differ** from local priors, whose class distribution is **independent**, but changes **continuously**, allowing for the observation of trends in its changes,
  - **DDIS** (*discrete dynamically imbalanced streams*)
    The global prior **may differ** from local priors, whose class distribution is **independent** and changes **discreetly**, making it impossible to observe trends in its changes.

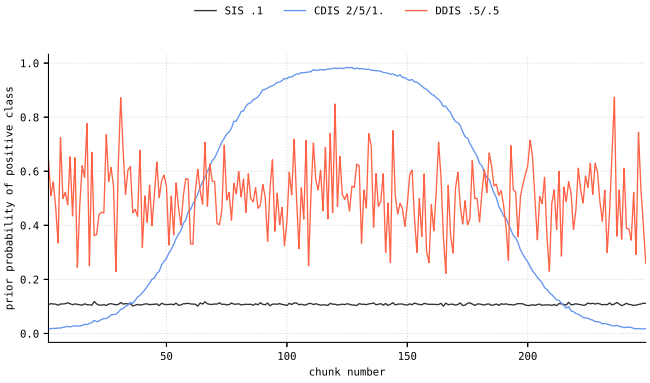

Fig. 1. Positive class *prior probabilities* for each data chunk of the streams from each category from introduced taxonomy.

The main contributions of this work are as follows:

- The proposition of a novel data stream taxonomy in the context of *prior class probability*.
- The proposition of an optimization method of *prior probability* estimation in dynamically imbalanced data streams from all four distinguished categories, based on multi-criteria regression using neural networks.
- Experimental evaluation of the proposed DSCA algorithm based on diverse data streams and a comparison with the *state-of-art* methods.

II. Dynamic Statistical Concept Analysis

This chapter describes the structure of *Dynamic Statistical Concept Analysis* (DSCA) model proposed in the work, which performs the *prior probability* prediction through *representation learning* [11].

Dynamic Statistical Concept Analysis is a novel method for determining *prior probabilities* in binary imbalanced data stream, designed primarily for DIS. It uses two *Multi-layer Perception (MLP) regressors* that are able to detect relationships between statistical characteristics of a data chunk, such as the *mean value* and *standard deviation* of the attributes in the context of processed chunk, and the class proportions.

The entire DSCA procedure is described in Algorithm 1. The model obtains information about the data chunk $\mathcal{DS}_k$ for the currently processed, k^{th} data chunk. Basic statistical characteristics is collected from each data portion – such as the *mean value* of the features and their *standard deviation*. This conglomerate contains approximated and reduced information about the entire batch, constructing r – a new *representation* of chunk.

Algorithm 1 DSCA pseudocode

Input:
Stream of data chunks $\{\mathcal{DS}_1, \mathcal{DS}_2, \ldots, \mathcal{DS}_k\}$,
Symbols:
$\mathcal{DS}_k$ – data chunk,
r – data chunk representation,
$\mathcal{R}$ – incremental chunk representation dataset,
P – real prior probability,
P' – predicted prior probability,
e – regressor epoch limit,
$\mathcal{C}_j^k$ – number of samples from class j in k^{th} chunk,
$\mathcal{C'}_j^k$ – predicted number of samples from class j in k^{th} chunk,
w – window width

$\mathcal{R} \leftarrow \varnothing$
$reg_0 \leftarrow$ cloned base regressor for negative class
$reg_1 \leftarrow$ cloned base regressor for positive class
for all data chunk $\mathcal{DS}_k \in Stream$ **do**
  $P_k \leftarrow \mathcal{C}_0^k \div (\mathcal{C}_0^k + \mathcal{C}_1^k)$ ▷ Real prior
  $r \leftarrow [mean(\mathcal{DS}_k), std(\mathcal{DS}_k)]$ ▷ Representation
  if $\mathcal{R} = \varnothing$ **then**
    $iter \leftarrow e$
  else
    $\mathcal{C'}_0^k \leftarrow reg_0(r)$ ▷ Estimated class count
    $\mathcal{C'}_1^k \leftarrow reg_1(r)$
    $P'_k \leftarrow \mathcal{C'}_0^k \div (\mathcal{C'}_0^k + \mathcal{C'}_1^k)$ ▷ Estimated prior
    $err \leftarrow |P_{k-1} - P'_k|$
    $iter \leftarrow e \times err$
  end if
  $\mathcal{R} \leftarrow \mathcal{R} + r$
  $R \leftarrow$ last w of $\mathcal{R}$
  $C \leftarrow$ last w of $\mathcal{C}$
  while $iter > 0$ **do**
    $reg_0 \leftarrow reg_0(R, C_0)$ ▷ Model update
    $reg_1 \leftarrow reg_1(R, C_1)$
    $iter \leftarrow iter - 1$
  end while
end for

For subsequent chunks of the data stream, (*i*) representation r extracted from the $\mathcal{DS}_k$ ($\mathcal{R}$), (*ii*) real and predicted *prior probabilities* (P and P') and (*iii*) real and predicted counts of samples from each of the problem classes (C and C') are stored. The model uses regressors, trained with the last

w *representations* R, and the corresponding class counts C^k respectively for negative and positive class. The window width w is set by the parameter.

The model works for streams processed with the *test-then-train* protocol, which means that for currently processed chunk $\mathcal{DS}_k$ a prior prediction P' was made in the previous step. Depending on the error err between the prediction and the true proportions between the classes, the number of regressor training iterations $iter$ is calculated – the bigger the err variable, the more epochs. Upper limit of iterations e is set by the parameter. For the first data chunk, regressors are trained to the maximum number of epochs. At the beginning of stream processing, errors will be larger, which will result in longer training time, however, after a certain amount of data processed, errors will decrease and therefore will the training time. Such dependency is presented in Figure 2.

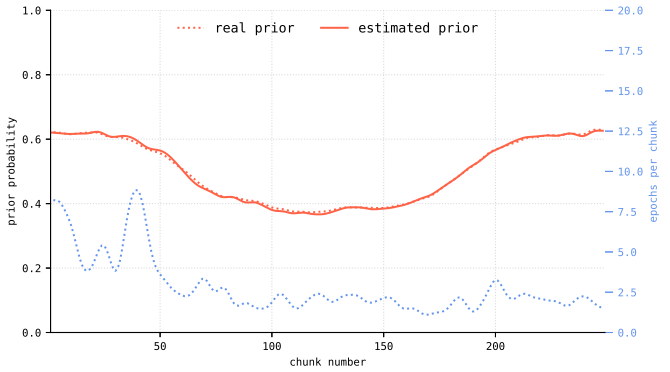

Fig. 2. Predicted and real *prior probability* for each data chunk of the stream with number of DSCA training epochs per given chunk.

During *prior probability* prediction procedure, described by lines 10-12 of Algorithm 1, where only feature values of the $\mathcal{DS}_k$ set are known. The responses of both regressors are used as predictions of individual classes count (C') in the current representation r, finally determining the predicted prior probability (P') as the proportions between the predicted number of the negative class samples and the predicted sum of the patterns in $\mathcal{DS}_k$.

III. Experiments design

The following section presents the research hypotheses, goals of the planned experiments, as well as the set-up of the conducted research.

a) Research hypotheses*:*

Using the taxonomy proposed in Section 1 and based on the observations from the works published in the field, we would like to verify the following experimental hypotheses:

- **H1** In the BS and SIS categories, it is possible to stabilize the information about the *prior probability* in each successive chunk by analyzing the distribution of classes in historical data (from the batches processed so far).
- **H2** In the CDIS category, it is possible to estimate the information about the *prior probability* in each successive chunk by analyzing the trend of the class distribution in historical data.
- **H3** In the DDIS category – due to the discrete changes in the class distribution – the estimators efficient in the remaining categories (BS, SIS and CDIS) will indicate low predictive ability.
- **H4** It is possible to propose a method of efficient prior probability estimation for the DDIS category.

In the experimental evaluation we will try to validate hypotheses H1-3 and verify the effectiveness of the proposed method in order to validate the hypothesis H4.

b) Goals of the experiments*:*

In order to verify the presented research hypotheses, two separate experiments were designed:

Experiment 1 – Hyperparameter optimization

The aim of the first experiment was to optimize hyperparameter n of the reference *prior probability* estimation methods, which denotes the number of previous data chunks, that are taken into consideration during the prior estimation process. The four parameterized methods were:

- **Linear** – methods dedicated to BS i SIS streams:
  - **Mean** – calculates the mean *prior probability* of n previous data chunks.
  - **Linear** – calculates the weighted mean *prior probability* of n previous data chunks. The newer the data chunk is, the more weight it is given.
- **Regressive** – methods dedicated to CDIS streams:
  - **Linear Regression (LR)** – with default hyperparametrization.
  - **Random Forest Regression (RF)** – with 100 trees.

Three n values were examined – consecutively 5, 50 and 250 (all) data chunks.

Experiment 2 – Comparative analysis

The aim of the second experiment was to compare the behavior of the proposed DSCA algorithm to the previously parameterized methods. One additional reference approach, called **Previous** is introduced, which assigns the current data chunk the same *prior probability* as in the previous one.

c) Experimental set-up*:*

To evaluate the proposed method performance, a total of 100 synthetic binary data streams with static concept were generated using the *stream-learn* package. Each data stream was composed of two hundred and fifty thousand instances (250 data chunks, 1000 instances each) described by 8 informative attributes and contained label noise at the level of 1%. During the generation process, 10 different stream types were distinguished, based on the (*a*) *prior class probability* taxonomy introduced in Section 1 and (*b*) class imbalance characteristics.

- **BS** (*balanced streams*) – a single data stream with balanced class distribution,

- **SIS** (*statically imbalanced streams*):
  - *the imbalance ratio* (IR) – successively 2.5, 5 and 10% of minority class,
- **CDIS** (*continuous dynamically imbalanced streams*):
  - *number of drifts* – 2,
  - *concept sigmoid spacing* – 5,
  - IR *amplitude* – successively 25, 50 and 100%,
- **DDIS** (*discrete dynamically imbalanced streams*):
  - *mean value* – 50%,
  - *standard deviation* – successively 10, 25 and 50%,

Additionally, each of the above-mentioned data streams was replicated 10 times with different *random states* to allow for a statistical analysis of obtained results using the *Student's t-test* [12].

As the conducted experiments deal with the *prior probability* estimation, the evaluation is based mainly on the *mean absolute error* (MAE) [13] regression loss calculated using *test-then-train* evaluation protocol [14]. Additionally, to facilitate the visual analysis of obtained results, figures presenting the accumulated difference between the actual and estimated *prior probability* value in each data chunk are presented. All experiments were implemented in the Python programming language, based on the *scikit-learn* [15] and *stream-learn* [16] API's, and can be replicated according to the code published on the *GitHub* repository[1].

IV. Experimental evaluation

This section presents the results of the conducted experiments. The tables present the results of a statistical analysis based on MAE and performed using *Student's t-test* ($p = .05$) on 10 stream replications. The numbers below the average MAE value indicate which of the n parameter values (Table I) or *prior probability* estimation methods (Table II) performed statistically significantly worse than the one in question.

A. Experiment 1 – Hyperparameter optimization

The results of *Experiment 1* are presented in Table I. It is noticeable that the effective value of *n* depends on the stream type:

- For *statically imbalanced streams* (SIS) and *balanced streams* (BS) the results present similar, due to the fact that the streams are stable, the greatest errors occur at the value of *n* equal to 1, which takes into account only previous data chunk. In case of these streams, 50 previous *prior probabilities* of data are sufficient to determine a stable *prior probability* level, hence no statistically significant improvement between *n* value equal to 50 and the analysis of all previous chunks.
- For *continuous dynamic imbalanced streams* (CDIS), the best results occurred for *n* equal to 1, where only proportions in classes of previous chunk were analyzed. This is due to the high instability of the stream while maintaining continuity. *Prior probability* in the analyzed chunk will be relatively close to the previous value. For RF method in streams with an amplitude of 25% and 50%, there was no statistically significant difference in the operation of the prediction.
- For *discrete dynamic imbalanced streams* (DDIS), the optimal *n*-value is the total number of chunks. Since the stream in the whole domain is stable and large fluctuations in *prior probabilities* occur between successive chunks, *n* equal to 1 will not operate well, and *n* equal to 50 still does not give a statistically better result than the analysis of all past *prior* values.

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

TABLE I
Results of Experiment 1: average mean absolute error for each of tested *n* value

	5 (1)	50 (2)	All (3)	5 (1)	50 (2)	All (3)
	BS			SIS .03		
Mean	0.014 —	0.013 1	0.013 1	0.005 —	0.005 —	0.005 —
Lin	0.014 —	0.013 1	0.013 1	0.005 —	0.005 —	0.005 —
LR	0.021 —	0.014 1	0.014 1	0.008 —	0.005 1	0.005 1
RF	0.016 —	0.016 —	0.016 —	0.005 —	0.006 —	0.005 —
	SIS .05			SIS .10		
Mean	0.006 —	0.006 —	0.006 1	0.009 —	0.008 1	0.008 1
Lin	0.007 —	0.006 1	0.006 1	0.009 —	0.008 1	0.008 1
LR	0.010 —	0.006 1	0.006 1	0.013 —	0.009 1	0.008 1
RF	0.007 —	0.007 —	0.007 —	0.010 —	0.010 —	0.009 —
	CDIS 2/5/.25			CDIS 2/5/.50		
Mean	0.015 all	0.050 3	0.084 —	0.017 all	0.097 3	0.167 —
Lin	0.015 all	0.036 3	0.070 —	0.016 all	0.067 3	0.138 —
LR	0.021 3	0.020 all	0.075 —	0.020 all	0.032 3	0.146 —
RF	0.016 —	0.016 —	0.016 —	0.015 —	0.015 —	0.015 —
	CDIS 2/5/1.			DDIS .5/.1		
Mean	0.024 all	0.192 3	0.332 —	0.086 —	0.080 1	0.079 all
Lin	0.019 all	0.131 3	0.274 —	0.088 —	0.080 1	0.080 all
LR	0.015 all	0.044 3	0.226 —	0.131 —	0.085 1	0.083 all
RF	0.014 all	0.015 —	0.015 —	0.096 —	0.096 —	0.095 all
	DDIS .5/.25			DDIS .5/.5		
Mean	0.209 —	0.194 1	0.193 all	0.330 —	0.312 1	0.311 all
Lin	0.212 —	0.194 1	0.193 all	0.334 —	0.313 1	0.312 1
LR	0.300 —	0.204 1	0.200 all	0.407 —	0.323 1	0.318 all
RF	0.231 —	0.231 —	0.229 all	0.357 —	0.357 —	0.354 all

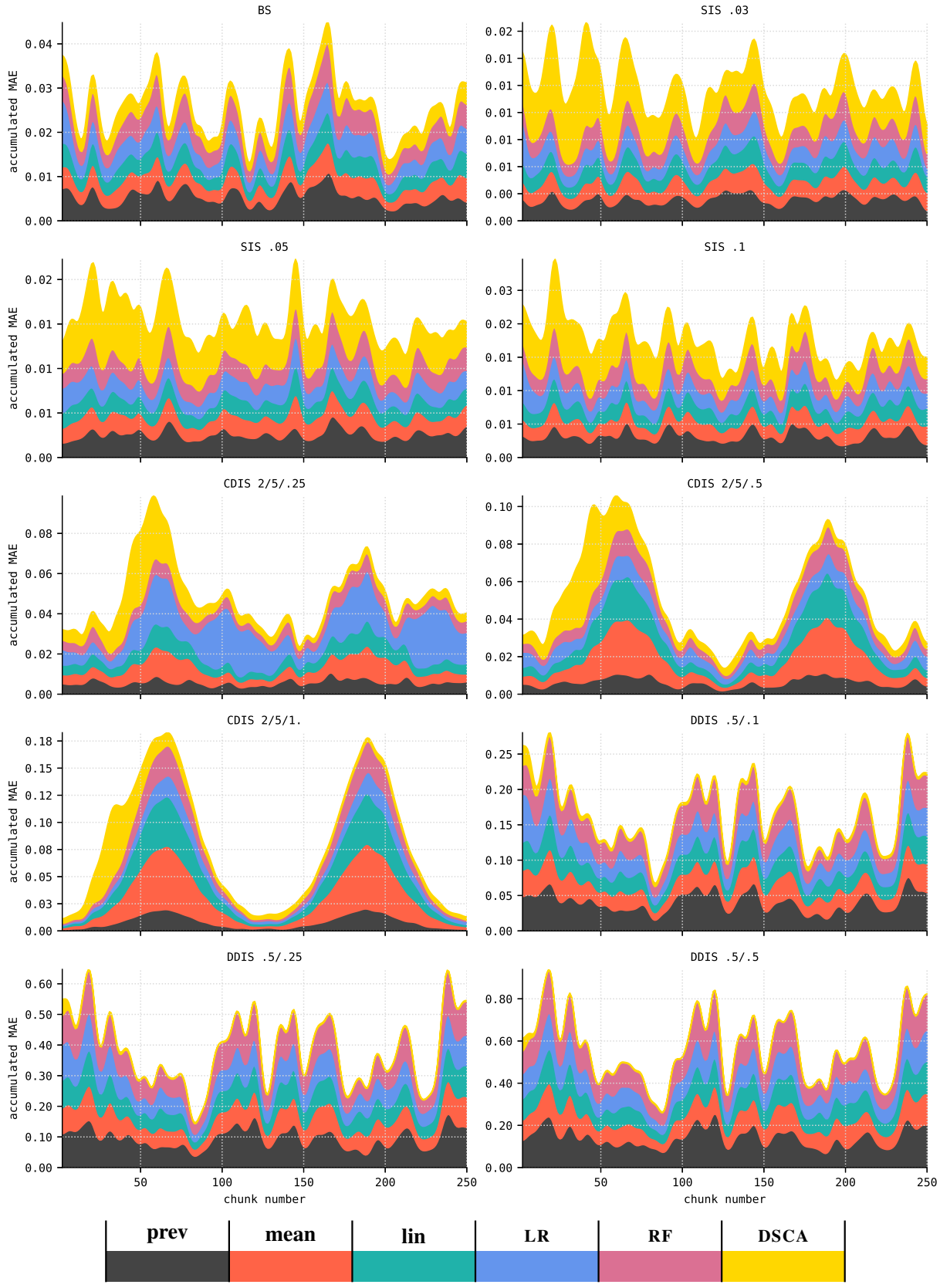

Fig. 3. Results of *Experiment 2* – accumulated difference between the estimated and actual *prior probability* value in each data chunk throughout stream analysis for all *prior probability* estimation methods and all examined stream types.

TABLE II
RESULTS OF EXPERIMENT 2: AVERAGE MEAN ABSOLUTE ERROR FOR EACH OF *prior probability* ESTIMATION METHODS

	LINEAR			REGRESSIVE		
	PREV (1)	MEAN (2)	LIN (3)	LR (4)	RF (5)	DSCA (6)
BS	0.018 —	0.013 1, 4, 5	0.013 1, 4, 5	0.014 1, 5	0.016 1	0.011 all
SIS .03	0.006 6	0.005 1, 5, 6	0.005 1, 5, 6	0.005 1, 5, 6	0.005 1, 6	0.010 —
SIS .05	0.008 6	0.006 1, 5, 6	0.006 1, 5, 6	0.006 1, 5, 6	0.007 1, 6	0.013 —
SIS .1	0.011 —	0.008 1, 5, 6	0.008 1, 5, 6	0.008 1, 5, 6	0.010 1	0.012 —
CDIS 2/5/.25	0.018 4	0.015 1, 4, 5	0.015 1, 4, 5	0.020 —	0.016 1, 4	0.012 all
CDIS 2/5/.5	0.017 2, 4	0.018 4	0.016 1, 2, 4	0.020 —	0.015 1, 2, 3, 4	0.012 all
CDIS 2/5/1.	0.013 2, 3, 4, 5	0.026 —	0.021 2	0.015 2, 3	0.015 2, 3, 4	0.013 2, 3, 4, 5
DDIS .5/.1	0.111 —	0.079 1, 4, 5	0.079 1, 4, 5	0.080 1, 5	0.094 1	0.010 all
DDIS .5/.25	0.267 —	0.192 1, 5	0.193 1, 5	0.193 1, 5	0.229 1	0.012 all
DDIS .5/.5	0.403 —	0.310 1, 5	0.310 1, 5	0.311 1, 5	0.354 1	0.013 all

B. Experiment 2 – Comparative analysis

Results of *Experiment 2* are presented in Figure 3. The graphs show the accumulated difference between the estimated and actual *prior probability* value in each data chunk throughout stream analysis for all *prior probability* estimation methods and all examined stream types.

It is noticeable, that the DSCA model learns best during a *prior probability* changes in subsequent chunks, therefore:

- For all *statically imbalanced streams* (SIS) the model does not perform better than simpler, linear methods.
- However, for *balanced streams* (BS), which are characterized by minor changes in *prior probabilities*, DCSA has higher prediction quality than for SIS. Predictive ability results from averaging two random *MLP regressor* models in DSCA that give a balanced prediction.
- For *continuous dynamic imbalanced streams* (CDIS) – the greater the changes in amplitude, the faster DSCA learns. Model's error at the beginning of stream processing is larger and decreases over subsequent data chunks. Such trend is also visible in Figure 2 and affects number of epochs of model's regressors training.
- For *discrete dynamic imbalanced streams* (DDIS) model converges almost instantly.

DSCA performs best for CDIS and DDIS, which seem to be the most difficult for other *prior probability* prediction methods. The final results of *Experiment 2* are also presented in Table II, extending the analysis with statistical tests. DSCA is statistically significantly better than all other tested methods for DDIS and two CDIS with amplitude of 25% and 50%. Results of CDIS with amplitude of 100% do not allow to select best method – DSCA and PREV performed equally well. However, it is worth noticing, that final scores of *prior probability* prediction methods were calculated over entire stream, whereas DSCA for CDIS needs more than 50 data chunks to converge and operate at the highest level.

V. CONCLUSIONS

The main purpose of this work was to introduce the original data stream taxonomy in the context of *prior class probability* as well as to propose a novel *prior probability* estimation method dedicated to *dynamically imbalanced streams* (DIS). This goal was achieved by introducing the *Dynamic Statistical Concept Analysis* (DSCA) model, which predicts the *prior probability* using *Multi-layer Perception regressors* to detect relationships between statistical characteristics of a data chunk and the number of class instances. Research conducted on the total of 100 data streams divided into 10 types in terms of *prior class probability* and class imbalance characteristics confirmed the usefulness of the proposed method, especially in the case of the *discrete dynamically imbalanced streams* (DDIS). Statistical analysis further confirmed the obtained results.

In connection with the performed statistical analysis, we can adopt (with the probability resulting from the statistical test parameters) the experimental hypotheses 1, 2 and 3, which results directly from the conducted observations. Experimental hypothesis 4 can be adopted for synthetic data with the presented DDIS characteristics.

Certain modifications to the DSCA model can be introduced to improve results at the start of stream processing, before the model is properly trained. Depending on the type of analyzed stream, up to the DSCA convergence point, the proportions of the classes from the previous data chunk can be taken as a prediction of *prior probability* — in case of CDIS, or the prediction of another method of determining *prior probability*, analyzing the previously processed chunks — in case of DDIS.

Further works will focus mainly on employing *prior probability* estimation methods in the dynamically imbalanced data stream classification task as well as on the *prior probability* estimation in multiclass imbalanced data streams.

ACKNOWLEDGMENT

This work was supported by the *Polish National Science Centre* under the grant No. 2017/27/B/ST6/01325.

REFERENCES

[1] W. Gibson, "Neuromancer," *Ace*, 1984.

[2] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView: IDC Analyze the future*, vol. 2007, no. 2012, pp. 1–16, 2012.

[3] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Information Fusion*, vol. 37, pp. 132 – 156, 2017.

[4] P. Domingos and G. 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.
[5] L. I. Kuncheva, "Classifier ensembles for changing environments," in *International Workshop on Multiple Classifier Systems*. Springer, 2004, pp. 1–15.
[6] J. Gao, B. Ding, W. Fan, J. Han, and S. Y. Philip, "Classifying data streams with skewed class distributions and concept drifts," *IEEE Internet Computing*, vol. 12, no. 6, pp. 37–49, 2008.
[7] H.-L. Nguyen, Y.-K. Woon, and W.-K. Ng, "A survey on data stream clustering and classification," *Knowledge and information systems*, vol. 45, no. 3, pp. 535–569, 2015.
[8] S. Wang, L. L. Minku, and X. Yao, "Resampling-based ensemble methods for online class imbalance learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1356–1368, May 2015.
[9] N. Rout, D. Mishra, and M. K. Mallick, "Handling imbalanced data: A survey," in *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications*. Springer, 2018, pp. 431–443.
[10] P. Ksieniewicz, "The prior probability in the batch classification of imbalanced data streams," *Neurocomputing*, Nov. 2020.
[11] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
[12] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.
[13] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.
[14] J. Gama, *Knowledge Discovery from Data Streams*, 1st ed. Chapman & Hall/CRC, 2010.
[15] 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, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[16] P. Ksieniewicz and P. Zyblewski, "stream-learn–open-source python library for difficult data stream batch analysis," *arXiv preprint arXiv:2001.11077*, 2020.