Employing One-class SVM Classifier Ensemble for Imbalanced Data Stream Classification

Jakub Klikowski[1][0000−0002−3825−5514]
and Micha Woniak[1][0000−0003−0146−4205]

Wrocaw University of Science and Technology,
{jakub.klikowski,michal.wozniak}@pwr.edu.pl

Abstract. The classification of imbalanced data streams is gaining more and more interest. However, apart from the problem that one of the class is not well represented, there are problems typical for data stream classification, such as limited resources, lack of access to the true labels and the possibility of occurrence of the *concept drift*. Possibility of *concept drift* appearing enforces design in the method adaptation mechanism. In this article, we propose the OCEIS classifier (*One-Class support vector machine classifier Ensemble for Imbalanced data Stream*). The main idea is to supply the committee with one-class classifiers trained on clustered data for each class separately. The results obtained from experiments carried out on synthetic and real data show that the proposed method achieves results at a similar level as the state of the art methods compared with it.

Keywords: One-class classification · Imbalanced data · Data streams · Ensemble learning

1 Introduction

Currently, the classification of difficult data is a frequently selected topic of research. One of many examples of this type of data is data streams. Such data should be processed for a limited time, having appropriate memory restrictions and performing only one-time use of incoming data. Also, the classifiers are required to be adaptable. A common phenomenon accompanying streams is the *concept drift*, which causes a change in the incoming data distribution. These changes may occur indefinitely.

Another problem is the imbalance of data, when it is combined with streams, significantly increases the difficulty. Uneven distribution of the number of classes is a fairly common phenomenon occurring in real data sets. This is not a problem when the differences are small, but it becomes serious when the difference between the number of objects from minority and majority classes is significantly huge. One of the known ways to deal with these difficulties is data sampling methods. These methods are designed to reduce the number of objects in the dominant class or to generate artificial objects of the minority class [2].

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_9

Designing methods with mechanisms for adapting to this type of data is another approach. One of this kind of approach is Learn++CDS [6] method, which combines the Learn++NSE [7] for nonstationary streams and SMOTE [2] for oversampling data. The next method in this paper is Learn++NIE, which is similar to the previous one, but with little difference. The classification error is introduced and some variation of *bagging* is used for balancing data. Wang et al. [19] design a method that uses the k-Mean clustering algorithm for undersampling data by prototype generation from centroids. The REA method proposed by Chen and He [4]. It is extension of the SERA [3] and the MuSeRA [5]. This family of methods uses a strategy for estimating similarity between previous samples of minority classes and the current minority data from the chunk.

One of the demanding situations when classifying imbalanced data streams is the temporary disappearance of the minority class or their appearance only in later stages. This type of phenomenon can cause a significant decrease in quality or sometimes prevent the typical classifier from working. The solution that raises this type of problem is the use of one-class classifiers that can make decisions based only on objects from one class only. Krawczyk et al. [11] proposed to the form an ensemble of one-class classifiers. Clustered data within samples from each class is used to train new models and expand ensemble. J. Liu et al. [14] designed a modular committee of single-class classifiers based on data density analysis. This is a similar approach, where clusters are created as part of a single-class data set. Krawczyk and Woniak [10] presented various metrics enabling the creation of effective one-class classifier committees.

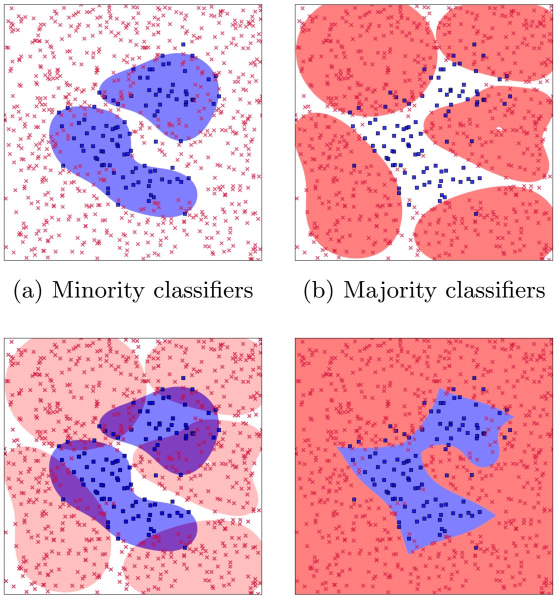
This paper proposes an ensemble method for classifying imbalanced data streams. The purpose of this work is to conduct preliminary experiments and analyze the obtained results, which will confirm whether the designed method can deal with imbalanced data streams competing in tests with the methods of state of the art. The main contributions of this work are as follows:

- A proposal for an OCEIS method for classifying imbalanced data streams based on one-class SVM classifiers
- Introduction of an appropriate combination rule allowing full use of the potential of the one-class SVM classifier ensemble
- Designing the proper learning procedure for the proposed method using division of data into classes and *k-mean* clustering
- Experimental evaluation of the proposed OCEIS method using real and synthetic generated imbalanced data streams and a comparison with the state-of-the-art methods

2 Proposed method

The proposed method **O**ne **C**lass support vector machine classifier **E**nsemble for **I**mbalanced data **S**tream (*OCEIS*) is a combination of different approaches to data classification. The main core of this idea is the use of one-class support vector machines (*OCSVM*) to classify imbalanced binary problems. This method is the chunk-based data stream method.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_9



(a) Minority classifiers (b) Majority classifiers

(c) Both classifiers (d) Ensemble

Fig. 1: Decision regions visualisation on the paw dataset from the Keel.es repository[1]

In the first step of the Algorithm 1, the chunk of training data is divided into a minority (D_{min}) and a majority set (D_{maj}). Then these sets of data are divided into clusters. Krawczyk et al. [11] indicate the importance of this idea. This decomposition of data over the feature space allows achieving less overlap of classifiers decision areas in the ensemble (Figure 1). The *k-means* algorithm [15] is used to create clusters. The key aspect is choosing the right number of clusters. Silhouette Value (SV) [18] comes with help, which allows calculating how similar an object is to its own cluster compared to other clusters. Kaufman et al. [9] introduced the Silhouette Coefficient (SC) for the maximum value of the mean SV over the entire dataset.

Minority and majority data is divided into clusters sets ($Cmin_{t,k}$, $Cmaj_{t,k}$) with a different number of centroids from 1 to K_{max}. The number of clusters with the highest value of SC is selected (K_{best}). This process is performed for minority and majority data. Then the formed clusters are used to fit new models ($h_{t,i}$, $h_{t,j}$) of OCSVM. These models are included in the pool of classifier committees (H_{min}, H_{maj}). The method is designed by default to operate on data streams. For this reason, a simple forgetting mechanism, also known as incremental learning, was implemented. This allows using models trained only on data with a certain time interval. When the algorithm reaches a set number (S) of chunks (t), in each iteration, the models built on the oldest chunk are removed from the ensemble.

A crucial component of any classifier ensemble is the combination rule, which makes decisions based on the predictions of the classifier ensemble. Designing a

DOI: 10.1007/978-3-030-50423-6_9

Algorithm 1 OCEIS - Train

```
Input:
    $D_t = \{(x_1^t, i_1^t), (x_2^t, i_2^t), (x_N^t, i_N^t)\}$ - training chunk of data stream
    $x_k^t \in \mathcal{X}$, where $\mathcal{X}$ stands for the feature space
    $i_k^t \in \mathcal{M} = \{minority, majority\}$, where $\mathcal{M}$ denotes set of the possible labels
    $t$ - current timestamp
    $N$ - chunk size
    $Dmaj_t$ - majority data chunk
    $Dmin_t$ - minority data chunk
    $OCSVM$ - SVM classifier for one-class classification
    $S$ - maximum size of classifier ensemble
    $K_{max}$ - maximum number of clusters
    $k$ - number of clusters
    $SilhouetteCoefficient$ - clusters consistency value [9]
    $K_{best}$ - number of clusters with best Silhouette Coefficient
    $KMeanClustering$ - k-mean clustering algorithm [15]
    $Cmaj_{t,k}$ - clusters of minority data $Dmaj_t$
    $Cmin_{t,k}$ - clusters of minority data $Dmin_t$
    $h_{t,j}$ - hypothesis from $OCSVM$ trained on $Cmaj_{t,j}$ cluster data
    $h_{t,i}$ - hypothesis from $OCSVM$ trained on $Cmin_{t,i}$ cluster data
    $H_{maj}$ - majority hypothesis set (ensemble)
    $H_{min}$ - minority hypothesis set (ensemble)

for $t = 1, 2, ...$ do
    Split $D_t$ into majority ($Dmaj_t$) and minority ($Dmin_t$) data
    for $k = 1, 2, ..., K_{max}$ do
        $Cmaj_{t,k} \leftarrow$ Call $KMeanClustering$ with $k$ on $Dmaj_t$
        $Cmin_{t,k} \leftarrow$ Call $KMeanClustering$ with $k$ on $Dmin_t$
    end for
    $K_{best} \leftarrow$ max Silhouette Coefficient on $Cmaj_{t,k}$
    for $i = 1, 2, ..., K_{best}$ do
        $h_{t,i} \leftarrow$ Call $OCSVM$ on $Cmaj_{t,i}$ cluster data
        Add $h_{t,i}$ to $H_{maj}$
    end for
    $K_{best} \leftarrow$ max Silhouette Coefficient on $Cmin_{t,k}$
    for $j = 1, 2, ..., K_{best}$ do
        $h_{t,j} \leftarrow$ Call $OCSVM$ on $Cmin_{t,i}$ cluster data
        Add $h_{t,j}$ to $H_{min}$
    end for
    if $t > S$ then
        Remove all $h_{t,i}$ where $t = t - S$ from $H_{maj}$
        Remove all $h_{t,j}$ where $t = t - S$ from $H_{min}$
    end if
end for
```

Algorithm 2 OCEIS - Prediction

Input:

$D_t = \{(x_1^t, i_1^t), (x_2^t, i_2^t), (x_N^t, i_N^t)\}$ - training chunk of data stream
$x_k^t \in \mathcal{X}$, where $\mathcal{X}$ stands for the feature space
$i_k^t \in \mathcal{M} = \{minority, majority\}$, where $\mathcal{M}$ denotes set of the possible labels
t - current timestamp
N - chunk size
$DecisionFunction$ - Signed distance to the separating hyperplane.
Returns positive value inside and negative outside hyperplane.
$Dist_{i,m}$ - distance from h_i decision boundary to x_m
$Dist_{j,m}$ - distance from h_j decision boundary to x_m
D_{maj} - maximum value of distance from h_j decision boundary to x_m
D_{min} - maximum value of distance from h_i decision boundary to x_m

```
for $t = 1, 2, ...$ do
    for each $h_j$ in $H_{maj}$ do
        $Dist_{j,m} \leftarrow$ Compute $DecisionFunction$ for $h_j$ on each $x_m$ in $D_t$
    end for
    for each $h_i$ in $H_{min}$ do
        $Dist_{i,m} \leftarrow$ Compute $DecisionFunction$ for $h_i$ on each $x_m$ in $D_t$
    end for
    for $m = 1, 2, ..., N$ do
        $D_{maj} \leftarrow$ max value of $Dist_{j,m}$ for $x_m$
        $D_{min} \leftarrow$ max value of $Dist_{i,m}$ for $x_m$
        if $D_{maj} > D_{min}$ then
            Predict majority class for $x_m$
        else
            Predict minority class for $x_m$
        end if
    end for
end for
```

good decision rule is vital for proper operation and obtaining satisfactory classification quality. First of all, OCEIS uses one-class classifiers and class clustering technique, which changes the way how the ensemble works. Well-known decision making based on majority voting [20] does not allow this kind of committee to make correct decisions. The number of classifiers for individual classes may vary significantly depending on the number of clusters. In this situation, there is a considerable risk that the decision will mainly base on majority classifiers.

OCEIS uses the original combination rule (Algorithm 2) based on distance from the decision boundary of classifiers to predicted samples. In the first step, the distances ($Dist_{i,m}$, $Dist_{j,m}$) are calculated from all objects of the predicted data to the hypersphere of the models forming the minority and the majority committee. The $DecisionFunction$ calculates these values. When the examined object is inside the checked hypersphere, it obtains a positive value, when it is outside, it receives a negative value. Then the highest value (D_{maj}, D_{min}) is determined from the majority and minority committees for each sample. When the best value (D_{maj}) for the model from the majority subensemble is greater than the best value (D_{min}) for the model from the minority subensemble, it means that this object belongs to the majority class. Similarly, when D_{min} is greater than D_{maj}, the object belongs to a minority class.

3 Experimental evaluation

The main purpose of this experiment was to check how good the proposed method performed with comparison to the other methods for classifying imbalanced data streams. The following research hypothesis was formulated:

> *It is possible to design a method with a statistically better or equal classification quality of imbalanced data streams compared to the selected state of the art methods.*

3.1 Experiment setup

All tests were carried out using 24 generated streams and 30 real streams (Table 1). The generated data comes from stream-learn [12] generator. These generated data differ in the level of imbalance: 10%, 20%, 30%. Label noise: 0% or 10% and type of drift: incremental or sudden. All generated data streams have 10 features, two classes and consist of 100,000 objects each. The proposed method has been tested with the selected state of the art methods:

- L++CDS [6]
- L++NIE [6]
- KMC [19]
- REA [4]
- OUSE [8]
- MLPC [16] (as a baseline)

The SVM implementation from the *scikit-learn* framework [17] was used as the base classifier in all committees. OCEIS implementation and the experimental environment is available on public github repository.[1] Four metrics were

[1] https://github.com/w4k2/oceis-iccs2020

used to measure the quality: Gmean, precision, recall and specificity. The results obtained in this way were compared using Wilcoxon statistical pair-tests. Each method was compared with OCEIS and these wins, lost and draw are shown in Figure 2 and Figure 3.

Table 1: Overview of real datasets used in experimental evaluation (KEEL [1] and PROMISE Software Engineering Repository [13]), IR - Imbalance Ratio

Dataset	IR	Samples	Features
abalone-17_vs_7-8-9-10	39	2338	8
australian	1.2	690	14
elecNormNew	1.4	45312	8
glass-0-1-2-3_vs_4-5-6	3.2	214	9
glass0	2.1	214	9
glass1	1.8	214	9
heart	1.2	270	13
jm1	5.5	2109	21
kc1	5.5	2109	21
kc2	3.9	522	21
kr-vs-k-three_vs_eleven	35	2935	6
kr-vs-k-zero-one_vs_draw	27	2901	6
page-blocks0	8.8	5472	10
pima	1.9	768	8
segment0	6	2308	19
shuttle-1vs4	14	1829	9
shuttle-1vsA	3.7	57999	9
shuttle-4-5vsA	3.8	57999	9
shuttle-4vsA	5.5	57999	9
shuttle-5vsA	17	57999	9
vehicle0	3.3	846	18
vowel0	10	988	13
wisconsin	1.9	683	9
yeast-0-2-5-6_vs_3-7-8-9	9.1	1004	8
yeast-0-2-5-7-9_vs_3-6-8	9.1	1004	8
yeast-0-3-5-9_vs_7-8	9.1	506	8
yeast-0-5-6-7-9_vs_4	9.4	528	8
yeast-2_vs_4	9.1	514	8
yeast1	2.5	1484	8
yeast3	8.1	1484	8

3.2 Results analysis

The obtained results of the Wilcoxon rank-sum pair statistical tests show that OCEIS can classify with the similar quality compared to the tested methods. For tested synthetic data streams (Figure 2) there is a certain advantage of the

L++CDS method over other methods. In second place can be put L++NIE and OCEIS. For the OUSE and L++NIE methods, there is a noticeable tendency to classify objects of the minority class, which is manifested by the higher results in the Recall (TPR) metric, but this causes a significant drop in Specifity (TNR). The worst in this test was the REA method, which shows a huge beat in the direction of the majority class. The results are more transparent for real data sets (Figure 3). Despite many ties, the best performing method is OCEIS. The exceptions are Recall for OUSE and Specifity for REA.

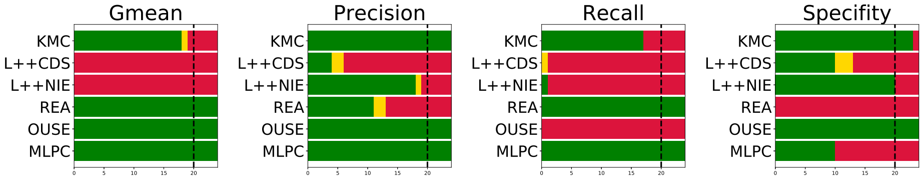


Fig. 2: Wilcoxon pair rank sum tests for synthetic data streams. Dashed vertical line is a critical value with a confidence level 0.05 (green - win, yellow - tie, red - lose)

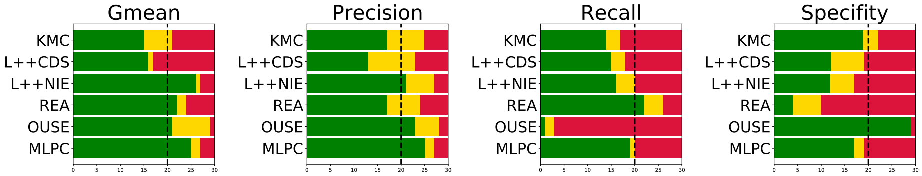


Fig. 3: Wilcoxon pair rank sum tests for real data streams. Dashed vertical line is a critical value with a confidence level 0.05 (green - win, yellow - tie, red - lose)

Charts of Gmean score over the data chunks provide some useful information about obtained results. To get a much better readability, the data before plotting was processed using a Gaussian filter. This procedure smoothes the edges of the results, which allows getting much more information from the results. The first observation is that for an incremental drift stream (Figure 4), OCEIS does not degrade quality over time. The negative effect of the concept drift can be seen on the KMC and REA methods, where the quality deteriorates significantly with the inflow of subsequent data chunks.

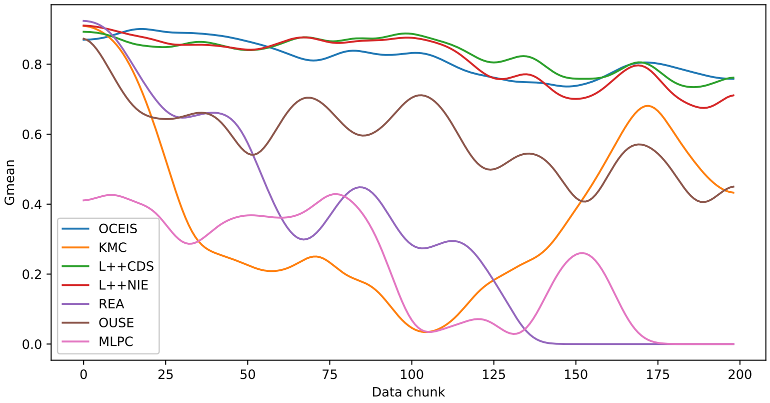


Fig. 4: Gmean score over the data chunks for synthetic data with incremental drift

In sudden *concept drift* (Figure 5), a certain decrease is noticeable, which is more or less reflected on every tested method. However, L++CDS, L++NIE and OCEIS can quickly rebuild this quality drop. This does not affect the overall quality of the classification significantly. Other methods perform a little bit randomly on sudden drifts. An example of the real-time shuttle-4vsA stream (Figure 6) shows the clear advantage of the OCEIS method over the other tested methods. A similar observation can be seen in other figures for real streams.

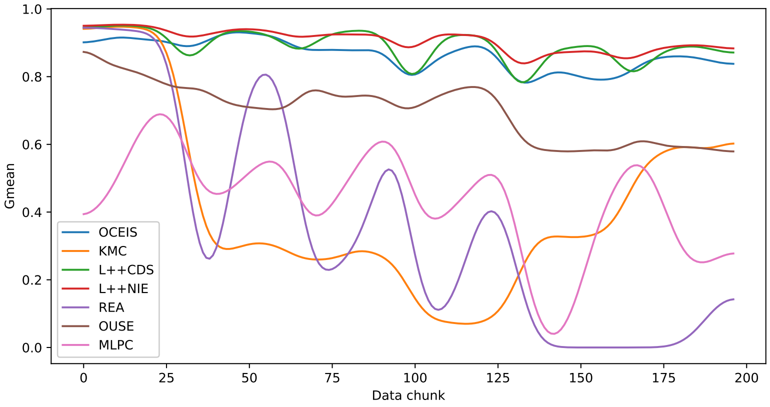


Fig. 5: Gmean score over the data chunks for synthetic data with sudden drift

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_9

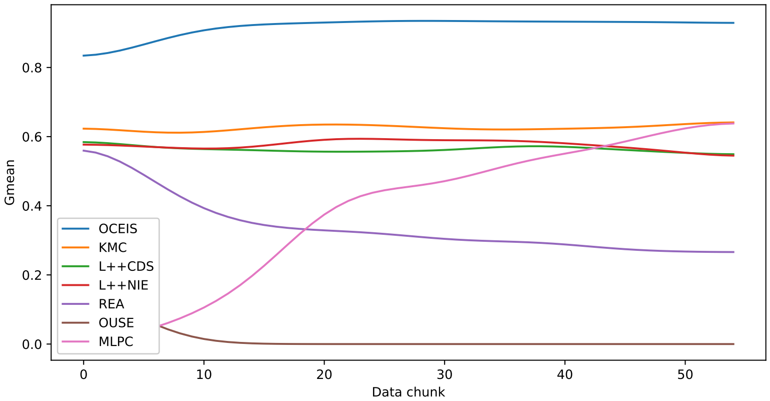


Fig. 6: Gmean score over the data chunks for real stream shuttle-4-5vsA

When analyzing the results, one should pay attention to the significant divergences in the performance of the proposed method for synthetic and real data streams. A large variety characterized real data streams, while artificial streams were generated using one type of generator (of course, for different settings). However, generated data streams are biased towards one type of data distribution, which probably was easy to analyze by some of the models, while the bias of the rest of them was not consistent with this type of data generator. Therefore, in the future, we are going to carry out the experimental research on the expanded pool of synthetic streams generated by other different generators.

4 Conclusions

We proposed an imbalanced data streams classification algorithm based on the one-class classifier ensemble. Based on the results obtained from reliable experiments, the formulated research hypothesis seems to be confirmed. OCEIS achieves results at a similar level to the compared methods, but it is worth noticing that it performs best on real stream data, which is its important advantage. Another advantage is that there is no tendency towards the excessive classification of objects from one of the classes. This was a problem in experiments carried out for the REA and OUSE methods. Such ”stability” contributes significantly to improving the quality of classification and obtaining satisfactory results.

For synthetic data streams, the proposed algorithm is not the worst-performing one. However, one can see some dominance of the methods from the Learn++ family, because the decision made by OCEIS is built based on all classifiers as part of the committee. One possible way to change this would be to break down newly created models by data chunks. This would build subcommittees (the Learn++NIE method works similarly). Then decisions would be made for each subcommittee separately. Expanding this by the weighted voting decision may significantly improve predictive performance. Another modernization of the method that would allow for some improvement would be the introduction of a drift detector. This mechanism would enable the ensemble to clean up after detecting *concept drift*.

The conducted research indicates the potential hidden in the presented method. It is worth considering extending the research to streams with other types of concept drifts. It is also beneficial to increase the number of real streams to test to get a broader spectrum of knowledge about how this method works on real data. One of the ideas for further research that arose while working on this paper is to test the operation on streams where the imbalance ratio changes over time. A very interesting would be an experiment on imbalanced data streams where the minority class temporarily disappears or appears after some time.

Acknowledgment

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

References

1. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic & Soft Computing **17** (2011)
2. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
3. Chen, S., He, H.: Sera: selectively recursive approach towards nonstationary imbalanced stream data mining. In: 2009 International Joint Conference on Neural Networks. pp. 522–529. IEEE (2009)
4. Chen, S., He, H.: Towards incremental learning of nonstationary imbalanced data stream: a multiple selectively recursive approach. Evolving Systems **2**(1), 35–50 (2011)
5. Chen, S., He, H., Li, K., Desai, S.: Musera: Multiple selectively recursive approach towards imbalanced stream data mining. In: The 2010 international joint conference on neural networks (IJCNN). pp. 1–8. IEEE (2010)
6. Ditzler, G., Polikar, R.: Incremental learning of concept drift from streaming imbalanced data. IEEE transactions on knowledge and data engineering **25**(10), 2283–2301 (2012)
7. Elwell, R., Polikar, R.: Incremental learning of concept drift in nonstationary environments. IEEE Transactions on Neural Networks **22**(10), 1517–1531 (2011)

DOI: 10.1007/978-3-030-50423-6_9

8. Gao, J., Ding, B., Fan, W., Han, J., Philip, S.Y.: Classifying data streams with skewed class distributions and concept drifts. IEEE Internet Computing **12**(6), 37–49 (2008)
9. Kaufmann, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. New York: Jonh Wiley (1990)
10. Krawczyk, B., Woźniak, M.: Diversity measures for one-class classifier ensembles. Neurocomputing **126**, 36–44 (2014)
11. Krawczyk, B., Woźniak, M., Cyganek, B.: Clustering-based ensembles for one-class classification. Information Sciences **264**, 182–195 (2014)
12. Ksieniewicz, P., Zyblewski, P.: stream-learn–open-source python library for difficult data stream batch analysis. arXiv preprint arXiv:2001.11077 (2020)
13. Lima, M., Valle, V., Costa, E., Lira, F., Gadelha, B.: Software engineering repositories: Expanding the promise database. In: Proceedings of the XXXIII Brazilian Symposium on Software Engineering. pp. 427–436. ACM (2019)
14. Liu, J., Miao, Q., Sun, Y., Song, J., Quan, Y.: Modular ensembles for one-class classification based on density analysis. Neurocomputing **171**, 262–276 (2016)
15. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 281–297. Oakland, CA, USA (1967)
16. Pal, S.K., Mitra, S.: Multilayer perceptron, fuzzy sets, classifiaction (1992)
17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
18. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)
19. Wang, Y., Zhang, Y., Wang, Y.: Mining data streams with skewed distribution by static classifier ensemble. In: Opportunities and challenges for next-generation applied intelligence, pp. 65–71. Springer (2009)
20. Xu, L., Krzyzak, A., Suen, C.Y.: Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE transactions on systems, man, and cybernetics **22**(3), 418–435 (1992)