Learning from Imbalanced Data Streams based on Over-Sampling and Instance Selection

Ireneusz Czarnowski[0000-0003-0867-3114]
Department of Information Systems, Gdynia Maritime University
Morska 83, 81-225 Gdynia, Poland
i.czarnowski@umg.edu.pl

Abstract. Learning from imbalanced data streams is one of the challenges for classification algorithms and learning classifiers. The goal of the paper is to propose and validate a new approach for learning from data streams. However, the paper references a problem of class-imbalanced data. In this paper, a hybrid approach for changing the class distribution towards a more balanced data using the over-sampling and instance selection techniques is discussed. The proposed approach assumes that classifiers are induced from incoming blocks of instances, called data chunks. These data chunks consist of incoming instances from different classes and a balance between them is obtained through the hybrid approach. These data chunks are next used to induce classifier ensembles. The proposed approach is validated experimentally using several selected benchmark datasets and the computational experiment results are presented and discussed. The results of the computational experiment show that the proposed approach for eliminating class imbalance in data streams can help increase the performance of online learning algorithms.

Keywords: classification, learning from data streams, imbalanced data, over-sampling, instance selection

1 Introduction

Data analysis is an area of intense research because data analysis is important from the perspective of potential business, medical, social, and industrial applications (see for example [1], [2], [3] and [4]). Much attention has been paid to data analysis because data analysis tools can be used to support decision-making processes. On the other hand, there are many real applications (including in human activities) which result in the growing volume of data as well as evolving their characteristics. Today, the commonly used term Big data emphasises both the aspect of data growth and the importance of data and its analysis, i.e. the volume and value of data are emphasized.

Big data is also defined in terms of velocity, which refers to data properties that are changed over time. It is another important dimension of the current data trend. In many real implementations, the data are accumulated with high speeds and flows in from different sources like, for example, machines, networks, social media, mobile phones, etc. Examples include Google or Facebook, on which 3.5 billion searches are made every day, and where the number of users has been seen to increase by approximately 22% from year to year. This also implies that there is a massive and continuous flow of data. Such

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32

a change of data properties over time is referred to as data drift, which has a relation with the dynamic character of the data source [5], [6]. Such changes are also referred to as a concept drift [7] or dataset shift [8]. Data streams are also referred to when new instances of the data are continuously provided.

When data analysis is a process of seeking out some important information in raw data as well as organising raw data to determine the usefulness of information, the process of extracting important patterns from such datasets is carried out under the umbrella of data mining tasks. Among these tasks is a classification, where assigning data into predefined classes is core. A classifier, which a role in the assigning of data to classes, is a model produced under the machine learning process. The aim of the process is to find a function that describes and distinguishes data classes. The process is called learning from data (learning from examples or, shortly, learning classifier from data) [9], where machine learning tools are used as learner models. From the implementation point of view, the process consists of a training and testing phase.

It should be noted that if a dataset with examples is categorical, then learning from examples is based on the existence of certain real-world concepts which might or might not be stable during the process of learning [9]. In the case of data streams, data which change over time together with the concept, involve changes which are difficult to predict in advance. Indeed, standard machine learning algorithms do not work with such data when their properties change over time. In other words, the algorithms cannot efficiently handle changes and the concept drift. Thus, learning classifiers from data streams are one of the recent important challenges in data mining. The challenge is to find the best way to automatically detect the concept drift and to adapt to the drift, as well as the associated algorithms, which will enable the fulfilment of all the data streaming requirements such as constraints of memory usage or restricted processing times. A review of different approaches for concept drift detection and their discussion is included in [14].

Data streams can be provided online, instance by instance or in block. It means that the learning algorithm can process instances appearing one by one over time, or in sets called data chunks. When the latter case, all learning processes are performed when all instances from the data chunk are available.

To deal with the limitations deriving from the stream character of data, so-called summarisation techniques can be used for data stream mining. Sampling or window models are proposed for such summarisations. From an implementation point of view, this means that a relatively small subset of data is processed. Of course, the size of the subset must be pre-set, but the core of such an approach is based on updating such a subset of data after a new chunk of data has arrived, and the removal of some instances from the subset with a given probability instead of periodically selecting them [10]. Other techniques are also available, including the weighted sampling [11] method or sampling within the sliding window model [12]. For example, the sliding window approach assumes that the analysis of the data stream is limited to the most recent instances. In a simple approach, sliding windows with a fixed size include the most recent instances and each new instance replaces the oldest instance. Window-based approaches are also useful due to their ability to quickly react to data changes. However, the size of the window is crucial. When the size is small the reaction can be relatively quick, although to the small size of the window may also lead to a loss of classification accuracy. Data summarisation techniques are also promising because they integrate well with drift detection techniques. Thus, when changes in the concept are detected a learner is updated or rebuilt. The main

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32

idea is based on keeping informative instances in the window frame, forgetting instances at a constant rate, and using only a window of the latest instances to train the classifier [13]. Of course, the question is which strategy should be implemented to the updating, including the forgetting, of the instances in the data window.

Another problem merging with learning from data streams concerns the process of classifier induction. A basic approach is based on so-called incremental learning. Incremental learning predicts a class label of the incoming instance and afterwards, information about whether the prediction was correct or not becomes available. The information may then be used for updating a classifier. However, a decision to modify or induce a new one classifier depends on the implemented adaptation mechanism. There are several different approaches to incrementally build new classifiers (see for example [13], [15]).

An approach supporting incremental learning from data streams can be also based on the decomposition of a multi-class classification problem into a finite number of the one-class classification problems [16]. This approach allows an independent analysis of the instances of each considered class, as well as the process of drift change monitoring. However, the results of such an independent analysis must be finally merged to obtain a classification model which will readily predict class labels for new instances following into the system.

Data stream mining requires the monitoring of drift detection as well as class distributions. Both problems belong to a challenging task itself. When the concept drift is detected it results in the data becoming unbalanced. Class imbalance is typical of streams of data and diametrically increases the difficulties associated with the learning of data process. Class imbalance may also negatively influence the learning process and decrease its accuracy. This phenomenon is important because these changes can be very dynamic. As has been underlined in [14], classes may switch roles, and minority classes may become the majority one, and vice versa. Such phenomena may have a dynamic character with a high frequency. The problem of class imbalance in learning from data streams is the main topic of this paper. This paper deals with a problem eliminating this phenomenon.

The aim of this work is to show that extending the functionality of the online learning approach, previously proposed in [17], by adding methods to balance the minority and majority classes, i.e., for supporting the analysis of the imbalanced data within the stream, increases the performance of the online learning algorithm. The proposed approach is based on over-sampling and under-sampling (i.e., instance selection) techniques that are implemented to form data chunks which are then used to induce the ensemble of classifiers. The main contribution of the paper is therefore to propose an over-sampling approach and extend the online learning framework presented in [17] through new computational functionalities.

This paper is organised as follows. The following section includes problem formulation. Then, a framework for learning from data streams is presented. In subsection 3.1, a proposed approach for changing the class distribution towards a more balance form are described; this subsection presents details of the proposed over-sampling approach. A detailed description of the computational experiment setup and the discussion of the experimental results is then included in section 4. Lastly, the final section contains conclusions and directions for future research.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32

2 Learning from Data Streams – Problem Formulation

A data stream can be considered as a sequence $\mathrm{X}_1, \ldots, \mathrm{X}_t, \ldots$, where X_i may be defined as a single instance or a set of instances, when instances appear not one by one in time but form sets, called data chunks. So, in the case of online learning from data streams, the sequence appears in a form $\{x_1, x_2, \ldots, x_t, \ldots\}$, where x_t is the *t*-th example's feature vector and *t* is a step of learning.

Considering the training phase, where the learner is produced, the sequence has a form of pairs $\{(x, c)_1, (x, c)_2, \ldots, (x, c)_t, \ldots\}$, where *c* is a class label associated with *x*, and is taken from a finite set of decision classes $C = \{c_i : i = 1, \ldots, d\}$. *d* is the number of classes established for the current step (the time step). During the training phase, such sequence pairs are provided to the learning algorithm as the training instances *T*. The role of the machine learning algorithm is to use these data pairs to find the best possible approximation *f'* of the unknown function *f* such that $f(x) = c$. After that, based on *f'* a class $c = f'(x) = c$ for *x*, where $(x, c) \notin T$ can be predicted.

In online learning, feedback on the actual label can be obtained after a prediction is made. In the case of an incorrect class assigned to *x* in step *t*, the feedback information can be used to update the function *f'* for the next steps.

When the classified data stream is given in a form of data chunks, it can be denoted by S_t meaning that it is the *t*-th data chunk. In such a case, the training set is formed from such data chunks ($S_t \subset T$ and $|S_t| < |T|$) or, of course, may form such a training set itself ($S_t = T$ and $|S_t| = |T|$).

Set *T* can be also noted as a sum of subsets to whom the instances from different classes belong, i.e., as $T = T^1 \cup T^2 \cup \ldots \cup T^d$. When set T is analysed with regards to the number of different classes to whom the instances forming this set belong, the problem of within-class imbalance can be observed. The problem of imbalanced data exists, when $\exists_{i,j \in \{1,\ldots,d\}} |T^i| \neq |T^j|$, where $i \neq j$. In such a case $T^i_{minority}$ $(i \in [1, \ldots, d])$ contains the minority class dataset, which means that the cardinality of $T^i_{minority}$ is smaller than the cardinality of each of the remaining subsets of *T* representing the remaining classes. On the other hand, among these remaining subsets, there is the majority class subset containing the majority class instances.

In this paper, it is assumed that a training set is formed by one data chunk, so it is a case of $(S_t = T)$.

3 An Approach for Learning from Imbalanced Data Streams

3.1 Online Classifier for Data Streams

In this paper, the problem of online learning from data streams is solved using the framework that had previously been proposed in [17]. The framework is based on three components which focus on data summarisation, learning, and classification.

The framework involves the processing of data chunks which consist of prototypes formed from a sequence of incoming instances for which predictions were incorrect. Data chunks are formed by the data summarisation component. This component consists of methods responsible for extracting instances incoming from a classification component.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32

The aim of the component is to permanently update data chunks by selecting adequate instances (called prototypes) from incoming data, memorising them and forgetting/removing other (non-informative) instances. The role of this component is to then pass the created or updated data chunk to the learning component. Such an updating mechanism is intended to help the system adapt to outside changes viewed in the data. Forming data chunks from incoming instances for which predictions were incorrect is also an approach for concept drift detection within data streams.

The idea implemented within the discussed approach is also based on the integration of online learning by sequentially inducing the prediction model. The existing feedback is assumed from a comparison of predicted class labels with the true labels. It means, in the proposed case, that a new classifier is constructed whenever a new data chunk becomes available.

The discussed framework is also based on a decomposition of the single multi-class classification problem into a set of one-class classification problems. This implies that a multi-class classification problem is solved using an ensemble of single one-class classifiers, i.e., one for each target class. It thus means that the incoming learning component data chunks are partitioned into subsets with so-called positive and negative instances for each considered decision class. Based on such a preparation of training data, a pool of the simple base classifiers is induced. These classifiers are represented by the matrix Φ consisting of $d \times \tau$ elements, i.e., K one-class classifiers, one for each target class. The approach is also based on the remembering of earlier-induced classifiers and τ represents the earlier steps with respect to data chunks that do not already exist in the system and which have been forgotten. This also means that the ensemble consists of the fixed-size set of classifiers, depending on the value of τ. The ensemble is updated when a new data chunk arrives. Then, a new induced classifier replaces the worst component in the ensemble. However, the process is associated with weights assigned to each of the base classifiers, based on the WAE approach (see [14]), where the value of weight increases if the classifier has been taking the correct decisions.

Finally, the aim of the classification component is to classify new incoming instances using ensemble classifiers. Because the classifier is constructed from K one-class classifiers, the prediction result produced by the ensemble classifier is determined through the weighted majority vote.

In summary, online learning from data streams based on the proposed framework is the following:

- three components are integrated with feedback from a comparison of predicted class labels with the true labels as a core to adapting the system to changes within data,
- the learning classifier is carried out using data chunks,
- a training data set consists of one data chunk,
- data chunks are formed from incoming instances for which former predictions were incorrect,
- a data chunk is updated to represent the current and representative instances,
- the learning classifiers are based on a decomposition of the single multi-class classification problem into a set of one-class classification problems,
- data classification is carried out using a weighted ensemble for one-class classification,

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32

- the system is based on updating existing ensemble classifiers according to new incoming data.

The structure of the framework is shown in Figure 1. The next subsection provides more details on the process forming the data chunks, which is core to this paper.

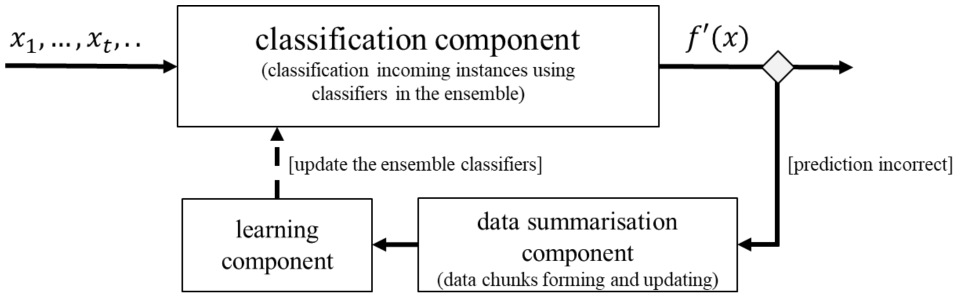

Fig. 1. Data processing by the proposed online learning approach

3.2 Training Data Set Forming and Updating

This subsection addresses the problem of forming data chunks from incoming instances. These data chunks are next used as a training data set. The proposed approach assumes that the size of the data chunk cannot be greater than the acceptable threshold. However, the following cases must be considered during the forming data chunks:

- the size of the data chunk is determined by the defined threshold, that respectively means that $\sum_{i=1}^{d} |T^i| = \alpha$, where α is the value of the threshold,
- when the size of the data chunk is smaller than the threshold size, incoming instances are being added to data chunk. However, each new instance is allocated to the corresponding subset of T, depending on the decision class of these instances,
- when the size of the data chunk is equal to the threshold, the chunk is updated and a new incoming instance replaces the other instance included in the current data chunk,
- the process of forming and updating the data chunk is guided in such a way as to maintain a balance between instances belonging to the considered classes, meaning that the sum of the sizes of the subsets for instances representing considered decision classes is not greater than the threshold,

The balance between instances belonging to minority and majority classes within data chunks is carried out using so-called data level methods. Data level methods aim to transform an imbalanced dataset into a better-balanced one by reducing the imbalance ratio between the majority and minority classes. The reduction can be carried out by over-sampling or under-sampling.

The aim of the under-sampling approaches is to balance the distribution of data classes. In practice, under-sampling techniques just remove instances from the majority class. The

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32

strength of this approach depends on the kind of rules that have been implemented for instance removal. Many methods belonging to this group are based on clustering and instance selection [19].

The proposed ensemble classifier for mining data streams assures a balanced distribution between minority and majority class instances using an approach based on instance selection. Instance selection aims to decide which from available and incoming instances should be finally retained and used during the learning process. So, the data chunk can be updated by replacing an older instance with a new one. Of course, the instance selection process can also decide whether or not to update the data chunk. The process of instance selection is carried out only on this part of instances belonging to the same decision class.

To decide whether the instances can be added to the current set, two well-known instance selection techniques, i.e., the Condensed Nearest Neighbour (CNN) algorithm and the Edited Nearest Neighbour (ENN) algorithm - both adopted to the considered one-class classification problem through applying the one-class k Nearest Neighbour method – called the Nearest Neighbour Description (NN-d) [20] – have been used. The adaptation assures instance selection processing for instances independently from considered decision classes. The pseudocode of the updating methods, denoted CNN-d and ENN-d respectively, are included in [17].

When in the current step of learning the number of instances in the subsets of T is not equal to the assumed threshold, the over-sampling procedure is activated on these subsets of instances to obtain a more balanced distribution of instances belonging to all classes.

The over-sampling procedure starts with identifying reference instances for two clusters of instances which do not belong to the minority class. In the presented clustering procedure, a k-means algorithm is used. The centres of the produced clusters are used for representing the reference instances. Next, for the reference instances, the procedure finds their neighbours belonging to the minority class. The closeness of neighbours is measured using the Euclidean measure and the number of neighbours is a parameter of the procedure. Next, an artificial instance located between the identified neighbours is generated randomly.

The pseudo-code explaining how an artificial instance is generated is shown as Algorithm 1. The algorithm is also illustrated in Figure 2. Algorithm 2 shows the pseudo-code of the proposed over-sampling procedure.

Algorithm 1 Generation of an artificial instance (GAI)

Input: x_1, x_2 – reference instances for the minority class; S – a subset of instances;
k - number of neighbours;
Output: x_a – an artificial instance;

Begin
For x_1 and x_2 find its k-nearest neighbour instances, which belong to S and where N contains the neighbour instances;
Generate randomly an artificial instance x_a located between instances from N;
Return x_a;
End

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32

Algorithm 2 Over-sampling procedure for the minority set of instances

Input: T - training set; k - number of neighbours; d - the number of classes; α is the value of the threshold
Output: $T = T^1 \cup T^2 \cup \ldots \cup T^d$ - sets of balanced instances forming a training set.

Begin
$\beta \cong \frac{\alpha}{d}$;
For i:=1,…,d **do**
If $|T^i| < \beta$ **then**
For subset $\bigcup_{j:j\in\{1,..,d\}\setminus\{i\}} T^j$ and run k-means algorithm for k=2 and for each obtained cluster return their centres as reference instances x_1^r and x_2^r;
$T^i = T^i \cup \{GAI(x_1^r, x_2^r, k, T^i)\}$;
End If
Return $T^1, T^2, \ldots, T^d$;
End

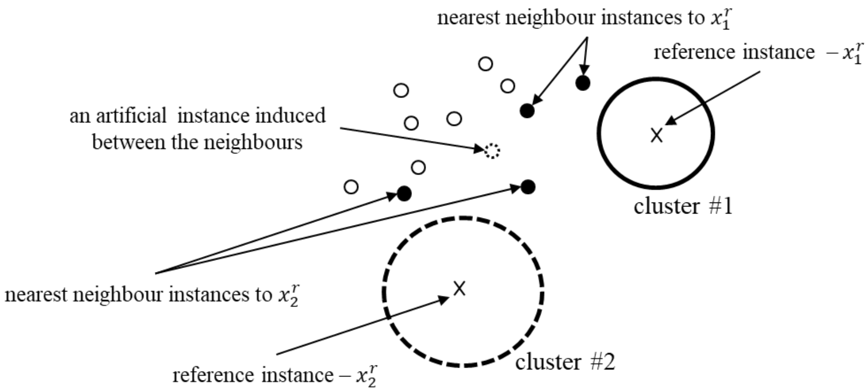

Fig. 2. Generation of an artificial instance

4 Computational Experiment

The computational experiment results are provided in this section. The aim of the experiment computation was to evaluate the performance of the approach discussed in this paper. The performance of the approach has been measured based on classification accuracy.

Based on this simple measure, the reported experiment aimed to answer the following question: whether the proposed approach, assuring a uniform class distribution in the

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32

training data set, is competitive with the other approaches dedicated to solving online learning from data streams.

In this paper, the proposed approach has been denoted as WECOI (Weighted Ensemble with one-class Classification and with Over-sampling and Instance selection). The WECOI approach has been implemented in two versions using the CNN-d and ENN-d algorithms for instance selection.

The WECOI has been also compared with its earlier versions denoted as WECU (Weighted Ensemble with one-class Classification based on data chunk Updating) and OLP (Online Learning based on Prototypes) (see [17]). The WECU has been implemented as an ensemble model updated based on the weights assigned to each one-class classifier. The WECU has also been implemented in a version based on a simple ensemble model with a simple majority voting to combine member decisions; such versions of the algorithm are denoted as WECUs. The OLP uses a simple ensemble model in which ensembles are updated by removing the oldest classifier. For comparison, WECOI has been also implemented in a version based on a simple ensemble model, denoted as WECOIs.

Both algorithms, i.e., WECU and OLP use CNN or ENN (or CNN-d and ENN-d in the case of WECU) to update data chunks but without balancing the class distribution within a training data set.

The aim has also been to compare the obtained results with others i.e., the Accuracy Weighted Ensemble (AWE) [23], the Hoeffding Option Tree (HOT), and the iOVFDT (Incrementally Optimized Very Fast Decision Tree) [21], [22], which are implemented as extensions of the Massive Online Analysis package within the WEKA environment. In the case of the WECOI and WECU approaches, the POSC4.5 algorithm has been used as a learning tool to induce base classifiers. In the case of OLP, the C4.5 algorithm has been applied to induce all the base models for ensemble classifiers.

The computational experiments have been carried out with parameter settings presented in Table 1. The values of the parameters have been set arbitrarily based on the trial-and-error procedure.

Table 1. Parameter setting in the reported computational resutls

Parameter	
τ – number of base classifiers	5
α – size of the data chunk	the value is shown in Table 2
number of neighbors for GAI	2
number of neighbors for ENN	3
metric distance for GAI	Euclidean
metric distance for ENN and CNN	Euclidean

All algorithms have been applied to solve the respective problems using several benchmark datasets obtained from the UCI Machine Learning Repository [24] and IDA repositories [25]. The used datasets are shown in Table 2.

An experiment plan was based on 30 repetitions of the proposed schema. The instances for the initial training set were selected randomly from each considered dataset. The number of selected instances was limited to the set threshold. The mean values of the

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32

classification accuracy obtained by the WECOI model based on an assumed experiment plan are shown in Table 3[1]. In the table the performances of WECU, OLP, AWE, HOT, as well as iOVFDT are also presented.

The results presented in Table 3 show that the WECOI model can be considered as a competitive algorithm. When the results are analysed for the approach used for instance selection, the general conclusion is that using ENN is superior to CNN. The observation holds true for all considered datasets.

Another observation is that using a simple majority voting does not guarantee competitive results like when using weighted majority votes.

Table 2. Characteristics of the dataset used in the experiment

Dataset	Source	#instances	#attributes	#classes	Best reported results classification accuracy	Threshold (as % of the data set)
Heart	UCI	303	13	2	83.8% !	10%
Diabetes	UCI	768	8	2	80.12% ‡	5%
WBC	UCI	699	9	2	99.3% ‡	5%
Australian credit (ACredit)	UCI	690	15	2	92.1% ‡	9%
German credit (GCredit)	UCI	1000	20	2	80.3% ‡	10%
Sonar	UCI	208	60	2	97.1% ″	10%
Satellite	UCI	6435	36	6	-	10%
Banana	IDA	5300	2	2	89.26% !	20%
Image	UCI	2310	18	2	80.3% †	20%
Thyroid	IDA	215	5	2	95.87% !	10%
Spambase	UCI	4610	57	2	82.37% ′	20%
Twonorm	IDA	7400	20	2	97.6% !	20%

Sources:
! - [26], ‡ - [27], † - [28], ″ - [24], ′ - [29]

[1] The best solution obtained by the compared algorithms is indicated in bold. The underline indicates the best solution obtained by the WECOI or WECU algorithm.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32

Table 3. Average classification accuracy (in %) obtained by the compared algorithms

Algorithm	Heart	Diabetes	WBC	ACredit	GCredit	Sonar	Satellite	Banana	Image	Thyroid	Spambase	Twonorm
WECOI-ENN	84,01	79,8	72,58	83,4	74,61	84,11	81,34	86,57	92,57	94,15	79,3	**97,73**
WECOIs-ENN	83,2	78,15	72,4	83,4	73,21	82,6	80,5	84,6	91,84	93,25	78,62	96,52
WECOI-CNN	80,81	78,15	71,89	82,06	74,03	82,32	80,78	87,02	91,8	94,06	77,86	96,43
WECOIs-CNN	81,24	77,58	71,64	81,52	73,25	82,2	79,1	86,4	91,42	93,52	77,05	95,62
WECU- ENN	**84,14**	79,62	72,54	83,74	**75,4**	**84,21**	79,14	88,1	91,47	94,01	78,5	97,7
WECUs-ENN	81,1	76,75	71,4	83,48	73,04	81,63	80,75	86,45	91,08	93,08	78,5	96,81
WECU- CNN	80,4	77,8	72,4	82,24	74,35	81,24	78,4	87,12	90,47	93,4	77,54	97,51
WECUs-CNN	81,24	76,57	72,11	80,84	73,91	82,2	76,54	86,02	91,21	91,47	76,34	94,24
OLP-ENN	80,4	72,82	71,21	82,4	71,84	75,44	78,25	85,72	90,07	91,47	78,17	96,4
OLP-CNN	78,14	73,2	70,1	81,52	70,06	76,81	80,45	84,1	90,32	93,14	77,61	95,1
AWE	78,01	72,5	**72,81**	84,5	73,5	77,02	82,4	87,4	91,61	93,1	75,4	76,8
HOT	81,4	**80,42**	72,67	82,41	72,87	76,05	**83,4**	86,77	92,2	94,21	77,4	96,06
iOVFDT	81,7	77,4	71,04	**84,5**	75,21	78,38	81,54	**89,21**	**95,07**	**94,63**	**80,2**	97

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32

WECOI can be also considered as competitive with the other algorithms for which results have also been obtained, i.e., AWE, HOT, and iOVFDT.

However, the main conclusion should be formulated for the results obtained by the approach aiming for a balance of instances distribution belonging to different decision classes. When analysing the results obtained by the WECOI and WECU models (results for these algorithms have been underlined in the table), better results were obtained with WECOI. Thus, the WECOI model is superior to the WECU one. On the other hand, both algorithms outperform OLP.

5 Conclusions

This paper contributes to our understanding of how data streams can work with imbalanced data, through extending the presented framework. Here, an approach for balancing class distributions within data chunks based on over-sampling and instance selection is proposed. These techniques have been implemented within the data summarisation component which aims to prepare training data to be used for learning classifiers. Over-sampling has been used as a tool for instance duplication in the minority class when instance selection has been used as a procedure for reducing the number of instances in the majority class.

The proposed approach has been evaluated and compared with other approaches and the computational experiment results show that the approach is competitive with others. Specifically, the proposed algorithm for balancing class distribution within the data stream overcomes its previous version where the imbalanced data problem was ignored.

Future research will focus on studying the influence of the size of data chunks as well as the number of neighbours in the generation of an artificial instance (GAI) on the accuracy of the proposed online learning approach. The future direction of the research will also allow the validation of different over-sampling techniques and the selection of the best one to be made.

References

[1] Kaplan, A. M., Haenlein, M.: Users of the world, unite! the challenges and opportunities of social media. Business Horizons 53(1), 59-68 (2010). doi:10.1016/j.bushor.2009.09.003

[2] Chan, J.F., Yuan, S., Kok, K., To, K.K., Chu, H., Yang, J., Yuen, K.: A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: A study of a family cluster. The Lancet, 395(10223), 514-523 (2020). doi:10.1016/S0140-6736(20)30154-9

[3] Phan, H. T., Nguyen, N. T., Tran, V. C., Hwang, D.: A sentiment analysis method of objects by integrating sentiments from tweets. Journal of Intelligent and Fuzzy Systems 37(6), 7251-7263 (2019). doi:10.3233/JIFS-179336

[4] Wang, Y., Zheng, L., Wang, Y.: Event-driven tool condition monitoring methodology considering tool life prediction based on industrial internet. Journal of Manufacturing Systems 58, 205-222 (2021). doi:10.1016/j.jmsy.2020.11.019

[5] Bifet, A.: Adaptive learning and mining for data streams and frequent patterns. PhD thesis, Universitat Politecnica de Catalunya (2009)

[6] Sahel, Z., Bouchachia, A., Gabrys, B., Rogers, P.: Adaptive mechanisms for classification problems with drifting data. In: Apolloni, B. et al. (eds.) KES 2007, LNAI 4693, pp. 419-426. Springer-Verlag Berlin Heidelberg (2007)

[7] Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23(1), 69-101 (1996)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32

[8] Tsymbal, A.: The problem of concept drift: definitions and related work. Tech. Rep. TCD-CS-2004-15, Department of Computer Science, Trinity College Dublin, Dublin, Ireland (2004)

[9] Mitchell, T.: Machine Learning. McGraw-Hill, New York (1997)

[10] Vitter, J.S.: Random sampling with a reservoir. ACM Transactions on Mathematical Software 11(1), 37–57 (1985)

[11] Chaudhuri, S., Motwani, R., Narasayya, V.R. On random sampling over joins. In: Delis, A., Faloutsos, C., Ghandeharizadeh, S. (eds.), SIGMOD 1999, pp. 263–274. ACM Press (1999)

[12] Guha, S., Mishra, N., Motwani, R., O'Callaghan, L.: Clustering data streams. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS), pp. 359–366, IEEE Computer Society, Washington (2000)

[13] Kuncheva. L.: Classifier ensembles for changing environments. In: Fabio Roli, F., Kittler, J. and Windeatt, T. (eds.), Multiple Classifier Systems, LNSC 3077, pp. 1–15, Springer, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-25966-4_1

[14] Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F.: Learning from imbalanced data streams, In: Learning from Imbalanced Data Sets. Springer, Cham. pp. 279–303 (2018). doi: 10.1007/978-3-319-98074-4_11

[15] Stefanowski, J.: Multiple and hybrid classifiers. In: Polkowski L. (ed.), Formal Methods and Intelligent Techniques in Control, Decision Making. Multimedia and Robotics, pp. 174-188, Warszawa (2001)

[16] Zhu, X., Ding, W., Yu, P.S.: One-class learning and concept summarization for data streams. Knowledge Information Systems 28:523:553 (2011)

[17] Czarnowski, I., Jędrzejowicz, P.: Ensemble online classifier based on the one-class Base classifiers for mining data streams. Cybernetic and Systems 46(1–2), 51–68 (2015). doi:10.1080/01969722.2015.1007736

[18] Wozniak, M., Cal, P., Cyganek, B.: The influence of a classifiers' diversity on the quality of weighted again ensemble. In: Nguyen, N.T., Attachoo, B., Trawinski, B., Somboonviwat, K. (Eds.) Intelligent Information and Database Systems - 6th Asian Conference, ACIIDS 2014, Bangkok, Thailand, 2014, LNAI 8398, pp. 90-99, Springer-Verlag Berlin Heidelberg (2014)

[19] Tsai, C-F., Lin, W-C., Hu, Y-H., Ya, G-T.: Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. Information Sciences 477, 47-54 (2019). doi: https://doi.org/10.1016/j.ins.2018.10.029

[20] Khan, S., Madden, M.G.: One-class classification: taxonomy of study and review of techniques. The Knowledge Engineering Review 29(3) 345 - 374 (2014)

[21] Bifet, A.: Adaptive learning and mining for data streams and frequent patterns. PhD thesis, Universitat Politecnica de Catalunya (2009)

[22] Bifet, A., Holmes, G., Kirkby, R., Pfahhringer, B.: MOA: Massive online analysis. Journal of Machine Learning Research 11, 1601-1604 (2010)

[23] Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: Proceedings of 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 226-235 (2003). https://doi.org/10.1145/956750.956778

[24] Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science (2007). http://www.ics.uci.edu/~mlearn/MLRepository.html

[25] IDA Benchmark Repository (2014) https://mldata.org/

[26] Wang, L., Hong-Bing, J., Jin, Y.: Fuzzy passive-aggressive classification: a robust and efficient algorithm for online classification problems, Information Sciences 220:46-63 (2013)

[27] Jędrzejowicz, J., Jędrzejowicz, P.: Rotation forrest with GEP-induced expression trees. In: Shea, J.O., et al. (eds.) Systems: Technologies and Applications, LNAI, vol. 6682, pp. 495-503, Springer- Heidelberg (2011)

[28] Jędrzejowicz, P., Jędrzejowicz, J.: A family of the online distance-based classifiers. In: Nguyen, N.T., Attachoo, B., Trawiński, B., Somboonviwat, K, (Eds.): Intelligent Information and Database Systems - 6th Asian Conference, ACIIDS 2014, Bangkok, Thailand, Proceedings, Part II. LNCS 8398, pp. 177-186, Springer (2014)

[29] Bertini, J.B., Zhao, L., Lopes, A.A.: An incremental learning algorithm based on the K-associated graph for non-stationary data classification. Information Sciences 246:52-68 (2013)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_32