# Ensemble Clustering for Novelty Detection in Data Streams

Kemilly Dearo Garcia[1,2](✉), Elaine Ribeiro de Faria[3], Cláudio Rebelo de Sá[1], João Mendes-Moreira[4], Charu C. Aggarwal[5], André C. P. L. F. de Carvalho[2], and Joost N. Kok[1]

[1] University of Twente, Enschede, The Netherlands
k.dearogarcia@utwente.nl
[2] University of São Paulo, São Paulo, Brazil
[3] Fed. University of Uberlandia, Uberlandia, Brazil
[4] LIAAD-INESC TEC, Faculty of Engineering, University of Porto, Porto, Portugal
[5] IBM T.J. Watson Research Center, Yorktown, USA

**Abstract.** In data streams new classes can appear over time due to changes in the data statistical distribution. Consequently, models can become outdated, which requires the use of incremental learning algorithms capable of detecting and learning the changes over time. However, when a single classification model is used for novelty detection, there is a risk that its bias may not be suitable for new data distributions. A solution could be the combination of several models into an ensemble. Besides, because models can only be updated when labeled data arrives, we propose two unsupervised ensemble approaches: one combining clustering partitions using the same clustering technique; and other using different clustering techniques. We compare the performance of the proposed methods with well known novelty detection algorithms. The methods were tested on datasets commonly used in the novelty detection literature. The experimental results show that proposed ensembles have competitive performance for novelty detection in data streams.

**Keywords:** Novelty detection · Ensembles · Clustering · Data streams

## 1 Introduction

In many real world scenarios, data continuously arrives at a high rate in a non stationary way, named *data streams*. As new data arrives, models previously induced can become outdated [6], causing predictive loss. In addition, due to the great amount of data generated, it is impossible to store it in the main memory, requiring the elimination of previous outdated data and online processing of incoming data [4]. In data streams, three types of changes can be found in the literature: concept drift [6], recurring concepts [2] and novel concepts [4]. Concept drift refers to changes in the statistical properties of the concept, such, i.e., a change in the stochastic process generating the data [6]. Recurring concepts are

P. Kralj Novak et al. (Eds.): DS 2019, LNAI 11828, pp. 460–470, 2019.
https://doi.org/10.1007/978-3-030-33778-0_34

a special type of concept drift in which concepts that appeared in the past may recur in the future [2]. Novelty concepts are patterns that were not present during the training of a classification model [4], which appear in the data stream.

Novelty detection is a machine learning task based on the identification of new concepts [5]. Several state of the art approaches [12] consider novelty detection as a binary classification task, composed by *normal* and *abnormal* classes. However, more recent approaches address novelty detection as a multi-class classification task [4]. The *abnormal* classes can be also named as *not normal* [12], *anomaly* [10] or *novel/new* [4] classes. We follow the notation from [4]. In the latter, normal concepts are a set of classes used to train the classification model and novelty concepts are the *new classes* that emerge over time.

In this work, we propose an ensemble of clustering partitions for novelty detection in data streams. We consider one ensemble obtained by a combination of different hyperparameter setting of the CluStream algorithm [1], referred as Homogeneous ensemble Clustering for data Streams (HoCluS). We also consider another ensemble with different clustering techniques, referred as Heterogeneous ensemble Clustering for data Streams (HeCluS). Each clustering technique can independently create and update a pre-defined number of partitions as new data arrive. This approach allows the use of clustering techniques with different bias, in order to obtain more robust classification models. In order to compare the performance of the different approaches, we implemented the two proposed methods in MINAS (MultI-class learNing Algorithm for data Streams) [4], a single classifier novelty detection algorithm for data streams. We conducted a set of experiments using datasets commonly referred in the novelty detection literature.

This paper is organized as follows. In Sect. 2 we present related work on novelty detection in data streams. In Sect. 3 we describe the proposed approaches and how they are incorporated into the MINAS algorithm. Section 4 presents the experiments performed. Finally, we conclude and discuss future research in Sect. 5.

## 2 Related Work

Several machine learning approaches have addressed novelty detection in data streams. Following, we describe the principal approaches according to two aspects: (i) number of classification models and (ii) strategy to update the classification model.

Considering the first aspect, we can divide the existing approaches in: single classification model or ensemble of models. Most of the single classification approaches use a $k$NN classification model based on a clustering approach [12]. This type of model can forget old clusters, insert new clusters and update the existing ones. Even though it is computationally less costly to train and update a single classification model, it may not be the most suitable to all time periods of a stream.

In contrast, other works focus on ensemble models. Ensemble classification for novelty detection in data streams are usually formed by combining classification models induced by the same algorithm [11]. In most approaches, the update of the ensemble consists of replacing the worst accurate model by a new one, obtained from the last labeled data chunk. However a disadvantage is that the ensemble in some cases has to wait for a long period before the labels are known. During this waiting period the predictive performance of the ensemble could decrease drastically.

Considering the second, the update strategy, these strategies are: *supervised* or *unsupervised*; according to presence/absence of labeled instances. Supervised approaches assume that the true label of all instances will eventually be available to update the model. Some examples of models using supervised learning are decision trees [11] and $k$NN [8].

On the other hand, unsupervised learning approaches assume that the true label will not be available. Therefore, they need to update the classification model without external feedback [5]. In general, they use the $k$-means algorithm to extract clusters to represent the current classes. As a result, they have some limitations: find only hyper-spherical clusters, have a fixed number of clusters and are sensitive to outliers.

## 3 Ensemble Clustering for Data Streams

The idea of Ensemble Clustering for Data Streams is similar to the general concept of combining classification models to construct an ensemble [13]. For that reason, it can be used to find the most suitable partition for a dataset. In this work, because we are constructing an ensemble of clustering, we are dividing the process in: the generation of a set of partitions and their combination into an ensemble of clustering.

Formally a data stream $D_{tr}$ is a potentially infinite sequence of instances arriving in a time $tr$, $tr \in \{1, ..., \infty\}$. Where, each instance, $X$, contains $d$ dimensions denoted by $X_{tr} = (X^1, ..., X^d)$ and a target class $y_{tr}$. A data stream can be represented as [4,7]: $D_{tr} = \{(X_1, y_1), (X_2, y_2), ..., (X_{tr}, y_{tr})\}$.

Novelty detection in data streams can be divided in two phases: the *offline* and the *online phase*. Assuming that in the offline phase a dataset has $m$ classes. Then, $Y^{Nor} = \{y_1, y_2, ..., y_m\}$ represents the set of Normal Classes. These class labels and the corresponding data samples are used to build the initial classification model. When during the online phase a novel class with label $y_{m+1}$ emerges, a novelty detection approach needs to detect this new class (concept) as quickly as possible and update the classification model accordingly.

### 3.1 The MINAS Algorithm

To test the proposed approaches, we implemented HoClus and HeCluS into the algorithm MINAS. The algorithm MINAS, in the offline phase, has a single model built with labeled data from the *normal* classes. This phase happens only once at the initial stage. The dataset with the labeled instances is split into subsets of data, each one containing data from one class in $Y^{Nor}$. Then, a clustering algorithm is applied on each subset to create a partition for each class. A cluster $\mathcal{C}_j$ is defined by a centroid $c_j$, a radius $r_j$ and a class label $y$. The radius of a cluster is the Euclidean distance between the centroid and the farthest data point in that cluster [11].

In the online phase the model calculates the Euclidean distance between each instance and the centroids of the clusters from the *normal* classes. If the smallest distance is less then the radius of the closest cluster, then the instance gets the label from that cluster. Otherwise, the instance is labeled as *Unknown* and stored in a buffer for future analysis. When the buffer is full of instances labeled as *Unknown*, a clustering algorithm is applied to obtain new clusters. A cluster is considered as *concept drift* if the distance between its centroid and the centroid from the nearest cluster from the *normal* class is bellow a given threshold. Otherwise, the cluster is considered a *novelty*. The instances that are not similar to any cluster, the outliers, are removed. Finally, the buffer is empty and this process is repeated every time the buffer is full.

## 3.2 Ensemble Clustering Applied to MINAS Algorithm

In this section we will explain how the two Ensemble Clustering for Data Streams were embedded in the MINAS algorithm. Both offline and online phase use two steps to build the ensemble of partitions: generation and combination. In the generation step, a user defined number of $P$ partitions from $N$ clustering techniques is generated, from the dataset $D_{tr}$. The output is an ensemble, $L_N$, containing $P \times N$ clustering partitions.

Given the ensemble $L_N$, we need to verify which clusters, from different partitions, are similar. For that the consensus function computes the Euclidean distance between the centroids of each cluster from the different partitions. If the distance between them is smaller than the sum of their radius, then those clusters will share the same label.

In the combination step, to use the partitions as an ensemble, it is necessary that similar clusters from different partitions have the same label. A cluster $\mathcal{C}_j$ with centroid $c_j$ is similar to a cluster $\mathcal{C}_k$ if the Euclidean distance between them is less than the sum of their radius: $EuclideanDistance(c_j, c_k) < (r_j + r_k)$.

In the offline phase, as in MINAS (See Sect. 3.1), the labeled instances are separated by labels in subsets. For each subset an Ensemble Clustering Generator is applied and $P$ partitions from $N$ clustering techniques are generated. In Fig. 1, the figures in gray represent MINAS algorithm. The ones with dashed lines indicate the parts that were adapted with the proposed method. Finally, the colored figures represent the steps of the Ensemble Clustering for Data Streams.

In the online phase, Fig. 2, the ensemble of clusters, blue diamond figure, using the majority vote of partitions, decides if a new instance is classified as *normal* or as *Unknown*. The instances classified as *Unknown* are stored in a buffer for future analysis. When the buffer reaches a given size $W$, new partitions are generated. This represented in Fig. 2 by circles/ellipses represent clusters and each color a label. After that, a consensus function, blue rectangle, is applied to combine all the clusters. The latter, will maintain only the clusters more similar

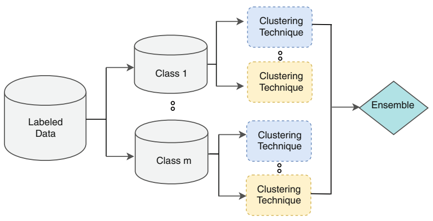


**Fig. 1.** Offline phase and ensemble clustering (adapted from [4])

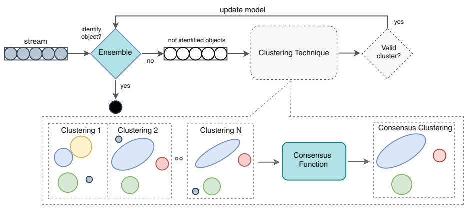


**Fig. 2.** Online phase and ensemble clustering (adapted from [4]) (Color figure online)

to other clusters from different partitions. Finally, these clusters are considered as novelties and then they are incorporated into the ensemble.

We define Homogeneous ensemble Clustering for data Streams (HoCluS) as an ensemble clustering obtained by the combination of $P$ partitions from the same algorithm. In this work we will use the algorithm for clustering in data streams CluStream [1]. CluStream is based on the $k$-means algorithm. Because of the random initialization phase of $k$-means, different partitions can be obtained. Because of that, an ensemble of CluStream partitions can be more robust for novelty detection than a single partition of CluStream.

We define a Heterogeneous Ensemble Clustering for data Streams (HeCluS) as the combination of $P$ partitions from $N$ different clustering techniques. In this work, our HeCluS has one model induced by each one of the following clustering algorithms for data streams: CluStream [1], DenStream [3] and ClusTree [9]. The main motivation to use DenStream, is because it is a stream clustering algorithm that is able to find clusters with arbitrary shape. Besides, it can also handle outliers [3]. On the other hand, we also use the ClusTree algorithm which also has a different bias from the other two. ClusTree builds clusters in a hierarchical data structure and can automatically set a number of clusters with arbitrary shape.

## 4 Experimental Setup and Results

We present in this section the experiments carried out for this study. We start by describing the datasets, then 2 we detail the experiment setup and finally we discuss the results.

The experiments were performed with synthetic and real datasets, both commonly used in novelty detection studies [1,3,5,11]. The synthetic datasets are: MOA, SynD, 1CDT, UG_2C_2D and Gear. The *MOA* dataset [4] has concept drift, appearance of new classes, recurrence and disappearance of existing classes. The clusters in this dataset are shaped as normally distributed hyper-spheres. The *SynD* [11] does not contain new classes, but does include concept drift. Finally, *1CDT, UG_2C_2D* and *Gear* are non stationary datasets[1]. In these datasets a single novelty occurs and concept drifts happen every 400 instances in 1CDT, 1000 instances in UG_2C_2D and 2000 instances in Gear. We also use two real datasets, *Forest Cover* and *KDD-CUP'99 NetWork Intrusion*[2]. The *KDD* dataset was used by [1] and [3].

We assume that the instances in the training data are the normal classes and new classes can appear during the online phase. In the offline phase, all methods are initialized with a batch of labeled data representing 10% of the data. We represent the results for each dataset with a confusion matrix (e.g. Table 1), which contains the percentage of: correctly classified classes (in gray), misclassified classes (in white), novelties detected (in gray) and *Unknown* instances (in white). The *Unknown* is the percentage of instances that the model was not able to classify. The sum of each column of the matrix should be 1. However, since we represent the average of 30 runs, the sum might not actually be 1. Whenever an instance is labelled as *Unknown*, it is considered as a classification error and counts as a *false negative*, which is a different approach adopted by [5]. For this reason, a high percentage of instances labelled as *Unknown*, will force the recall to be lower. This will make the comparison of the models more fair. The *F-Measure* is the weighted harmonic mean of precision and recall [5].

For datasets with more than two classes we used graphics to analyse the predictive performance of MINAS, HoCluS and HeCluS over time. We computed the F-Measure and *Unknown* every 10.000 instances. The algorithms CLAM, MINER and SAND, are not used in this analysis due difficulties to obtain the information necessary to calculate the F-Measure over time.

## 4.1 Results

We compare the predictive performance of HoCluS and HeCluS with the original MINAS and with three other supervised novelty detection methods: Miner [11], CLAM [2] and SAND [8]. For simplicity, we use the default hyper-parameters of the existing algorithms. In **Gear** dataset C1 and C2 are *normal* classes, both with concept drift. In Table 1 we can observe that the unsupervised method HeCluS has the highest *F-Measure*. Moreover, this was the only unsupervised method that does not misclassified C2 as a novelty. This can be due to the fact that HeCluS is able to build models with non-spherical clusters, which can better represent the classes of this dataset. We note that HoCluS and MINAS misclassify some instances as *novelties*, however this is less evident with HoCluS

[1] https://www.sites.google.com/site/nonstationaryarchive/.
[2] http://archive.ics.uci.edu/ml/index.php.

**Table 1.** Confusion matrix for gear dataset

| | Supervised | | | | | | Unsupervised | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SAND | | CLAM | | MINER | | MINAS | | HoCluS | | HeCluS | |
| | C1 | C2 | C1 | C2 | C1 | C2 | C1 | C2 | C1 | C2 | C1 | C2 |
| C 1 | 1.00 | 0.00 | 1.00 | 0.25 | 0.97 | 0.04 | 0.82 | 0.03 | 0.88 | 0.06 | 0.94 | 0.06 |
| C 2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 | 0.96 | 0.08 | 0.90 | 0.06 | 0.86 | 0.06 | 0.94 |
| Novelty | 0.00 | 0.57 | 0.00 | 0.00 | 0.00 | 0.00 | 0.10 | 0.07 | 0.02 | 0.03 | 0.00 | 0.00 |
| Unknown | 0.00 | 0.43 | 0.00 | 0.75 | 0.00 | 0.00 | 0.00 | 0.00 | 0.04 | 0.03 | 0.02 | 0.03 |
| F-Measure | 0.50 | | 0.80 | | 0.98 | | 0.92 | | 0.93 | | 0.97 | |

than with MINAS. In terms of the supervised methods, MINER has the best *F-Measure* with few misclassification. However, SAND and CLAM misclassify the majority data from C2 as *Unknown* or as C1. SAND classifies the drifts from C2 as *novelties* and CLAM does not learn from the *Unknown*.

In the **SynD** dataset C1 and C2 are the normal classes and both have concept drift. We observe, Table 2, that the methods show a similar behavior on the Gear dataset. In the group of unsupervised methods, HeCluS and HoCluS obtained a slightly better performance than MINAS. Considering the supervised methods, SAND, CLAM and MINER learn part of the concept drifts. SAND has *F-Measure* 0.77, however it has the highest percentage of *Unknown*. CLAM does not considered any instance as *Unknown*, but has more classification errors. MINER had the highest score and does not classify any instance as *Unknown*.

**Table 2.** Confusion matrix for Synd dataset

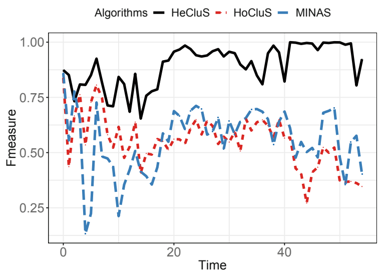
| | Supervised | | | | | | Unsupervised | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SAND | | CLAM | | MINER | | MINAS | | HoCluS | | HeCluS | |
| | C1 | C2 | C1 | C2 | C1 | C2 | C1 | C2 | C1 | C2 | C1 | C2 |
| C 1 | 0.64 | 0.00 | 0.50 | 0.45 | 0.88 | 0.18 | 0.63 | 0.34 | 0.66 | 0.31 | 0.69 | 0.34 |
| C 2 | 0.00 | 0.60 | 0.50 | 0.55 | 0.12 | 0.82 | 0.37 | 0.66 | 0.30 | 0.70 | 0.32 | 0.65 |
| Novelty | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Unknown | 0.36 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.01 | 0.01 | 0.01 | 0.00 | 0.00 |
| F-Measure | 0.77 | | 0.69 | | 0.92 | | 0.76 | | 0.78 | | 0.79 | |

The **1CDT** dataset, Table 3, has a normal class (C1) and a new class (C2) with concept drift. With this dataset we can evaluate the performance of the models with regard to novel concept detection in the online phase. We note that MINAS, HoCluS and HeCluS even during the online phase, they are not informed of the true class of the novelties detected. Because of that, even though they detect C2 as a novel class, their predictions are only represented as *novelty*. All models correctly classify most data from C1. On the other hand, in terms of the novel class, C2, they have different predictive performance. In terms of unsupervised approaches, HeCluS does not distinguish very well the normal concept, C1, from the novel concept, C2. On the other hand, HoCluS and MINAS, never misclassified C1 as C2. However, both have the highest percentage of *Unknown*. In terms of supervised approaches, SAND presents the lowest *F-Measure* because misclassifies most data from C2 as C1. SAND gives a confidence score for each model depending on their previous performance. Because of that the models representing the normal class have higher score than the novelty. This might explain the high percentage of misclassification of C2. The methods CLAM and MINER show high performance for C2, combining the percentage of correct classification and *novelty*.

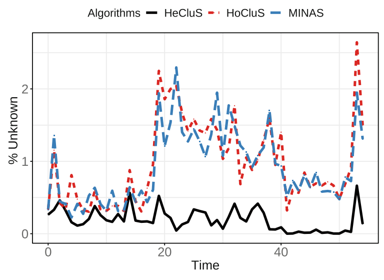
**Table 3.** Confusion matrix for 1CDT dataset

| | Supervised | | | | | | Unsupervised | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SAND | | CLAM | | MINER | | MINAS | | HoCluS | | HeCluS | |
| | C1 | C2 | C1 | C2 | C1 | C2 | C1 | C2 | C1 | C2 | C1 | C2 |
| C 1 | 1.00 | 0.84 | 1.00 | 0.00 | 1.00 | 0.00 | 0.94 | 0.00 | 0.94 | 0.00 | 1.00 | 0.20 |
| C 2 | 0.00 | 0.03 | 0.00 | 0.73 | 0.00 | 0.71 | 0.00 | 0.00 | 0.09 | 0.00 | 0.00 | 0.00 |
| Novelty | 0.00 | 0.11 | 0.00 | 0.27 | 0.00 | 0.27 | 0.04 | 0.87 | 0.00 | 0.87 | 0.00 | 0.75 |
| Unknown | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 | 0.02 | 0.06 | 0.12 | 0.00 | 0.13 | 0.00 | 0.06 |
| F-Measure | 0.62 | | 0.92 | | 0.99 | | 0.95 | | 0.95 | | 0.93 | |

In the **Forest Cover** dataset, we have 3 normal classes and 5 novel classes. We can observe in Fig. 3(a) that HeCluS has the highest *F-Measure* over time and it is more stable. In terms of *Unknown* data, Fig. 3(b), HoCluS and MINAS have higher percentage of data classified as *Unknown* than HeCluS, specially from time 20 to 40. In terms of F-Measure, SAND has performance of 0.32 due confusion between the normal and novel classes. Possibly this happens because of the confidence factor, a hyperparamenter, that tends to privilege majority classes, which is the case for the classes from the *normal* class. CLAM had performance of 0.72 because it was able to detect most of the *novel* classes. Finally, MINER 0.59 detected a high number of class, but misclassify the novelties as the normal classes.



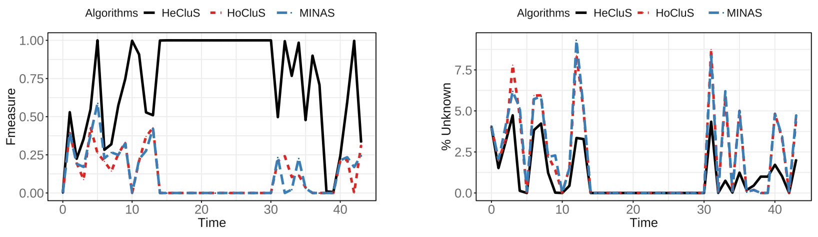
(a) F-Measure



(b) Unknown

**Fig. 3.** F-measure and *Unknown* for forest cover dataset

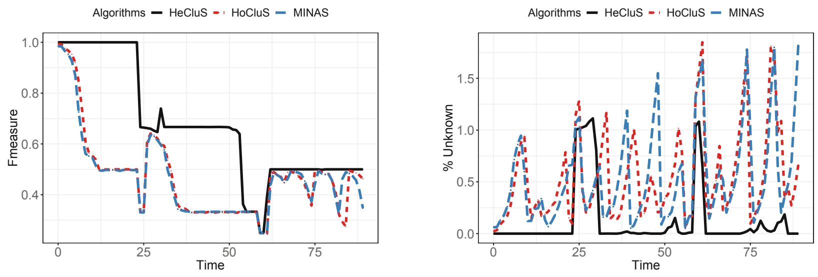
In the **KDD** dataset we consider 18 *normal* classes and 5 *novel* classes. All methods start in time 0 with a low *F-Measure*, Fig. 4(a). During the time 15 to 30, HeCluS has better performance and is more stable, but after that period shows great instability. MINAS and HoCluS have less instability, but with low performance and periods with not correctly classify classes. HoCluS and MINAS have higher *Unknown* data, Fig. 4(b). However, the low *F-Measure* and *Unknown* show that MINAS and HoCluS misclassify the classes. In terms of the final *F-Measure*, SAND has performance of 0.27, CLAM 0.32 and MINER 0.30.



(a) F-Measure (b) Unknown

**Fig. 4.** F-measure and unknown for KDD99 dataset

The **MOA** dataset has 2 *normal* classes and 2 *novel* classes. The normal classes have concept drifts from time 0 to 90 and from time 30 to 55 they overlap. The first new class emerge at time 35 and second new class emerge after time 75. In Fig. 5(a), we can see that HeCluS is better and more stable than HoCluS and MINAS. In terms of performance, HoCluS is not different then MINAS. However, observing the peaks in Fig. 5(b), we can see that HeCluS updated less times that HoCluS and MINAS. In that case, MINAS needed to update more times than HoCluS. In terms of F-Measure, SAND had performance of 0.88, CLAM 0.44 and MINER 0.90.



(a) F-measure (b) Unknown

**Fig. 5.** F-measure and Unknown for MOA dataset

The **UG_2C_2D** dataset has a *normal* class and a *new* class, both with concept drift and overlap. Analysing the unsupervised methods in Table 4, HeCluS classifies all data as C1 and does not learn the new class. HoCluS has slightly higher *F-Measure* than MINAS, both methods misclassify great percentage of C2 and C1, however they learn the new class. For the supervised methods, SAND and MINER also misclassify C2 as C1. MINER misclassify more C1 as C2 than the others methods. CLAM presents the best performance because does not misclassifies C2 as C1.

**Table 4.** Confusion matrix for UG_2C_2D dataset

| | Supervised | | | | | | Unsupervised | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SAND | | CLAM | | MINER | | MINAS | | HoCluS | | HeCluS | |
| | C1 | C2 | C1 | C2 | C1 | C2 | C1 | C2 | C1 | C2 | C1 | C2 |
| C 1 | 0.71 | 0.50 | 1.00 | 0.02 | 0.50 | 0.58 | 0.42 | 0.45 | 0.45 | 0.50 | 1.00 | 1.00 |
| C 2 | 0.10 | 0.07 | 0.00 | 0.70 | 0.46 | 0.37 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Novelty | 0.28 | 0.44 | 0.00 | 0.00 | 0.01 | 0.01 | 0.51 | 0.46 | 0.50 | 0.43 | 0.00 | 0.00 |
| Unknown | 0.01 | 0.06 | 0.00 | 0.28 | 0.03 | 0.04 | 0.07 | 0.09 | 0.05 | 0.07 | 0.00 | 0.00 |
| F-Measure | 0.72 | | 0.91 | | 0.60 | | 0.43 | | 0.45 | | 0.50 | |

## 5 Conclusions

In this work we proposed the methods HeCluS and HoCluS for detection of novelties and concept drift in data streams. These ensembles combine several partitions from one or more clustering techniques. This allows the use of clustering techniques with different bias at the same time, in order to obtain more robust classification models.

In the experiment with the datasets with only concept drift, we demonstrated that HoCluS and HeCluS are competitive with state of the art supervised methods. Observing the performance of HoCluS and HeCluS over time, we conclude that HeCluS has lower percentage of *Unknown* instances. This shows that HeCluS takes more risks in the classification decision than HoCluS. Because of that HeCluS was better in most datasets. On the other hand, this behavior also gives the model less chances to be updated.

The use of ensembles with different clustering techniques is a promising strategy, because the inductive bias of each classification model can be more suitable for a given data stream or only for during certain periods in the same data stream. The experiments showed that during the online phase, the performance of all tested algorithms were affected by the changes in the data distribution.

## References

1. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: VLDB 2003, Proceedings of 29th International Conference on Very Large Data Bases, 9–12 September 2003, Berlin, Germany, pp. 81–92 (2003)
2. Al-Khateeb, T., Masud, M.M., Khan, L., Aggarwal, C.C., Han, J., Thuraisingham, B.M.: Stream classification with recurring and novel class detection using class-based ensemble. In: 12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, 10–13 December 2012, pp. 31–40 (2012)

3. Cao, F., Ester, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: Proceedings of the Sixth SIAM International Conference on Data Mining, 20–22 April 2006, Bethesda, MD, USA, pp. 328–339 (2006)
4. Faria, E.R., Gama, J., Carvalho, A.C.P.L.F.: Novelty detection algorithm for data streams multi-class problems. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC 2013, Coimbra, Portugal, 18–22 March 2013, pp. 795–800 (2013)
5. Faria, E.R., Gonçalves, I.J.C.R., de Carvalho, A.C.P.L.F., Gama, J.: Novelty detection in data streams. Artif. Intell. Rev. **45**(2), 235–269 (2016)
6. Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM Comput. Surv. **46**(4), 44:1–44:37 (2014)
7. Garcia, K.D., de Carvalho, A.C.P.L.F., Mendes-Moreira, J.: A cluster-based prototype reduction for online classification. In: Yin, H., Camacho, D., Novais, P., Tallón-Ballesteros, A.J. (eds.) IDEAL 2018. LNCS, vol. 11314, pp. 603–610. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03493-1_63
8. Haque, A., Khan, L., Baron, M.: Semi supervised adaptive framework for classifying evolving data stream. In: Cao, T., Lim, E.-P., Zhou, Z.-H., Ho, T.-B., Cheung, D., Motoda, H. (eds.) PAKDD 2015. LNCS (LNAI), vol. 9078, pp. 383–394. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18032-8_30
9. Kranen, P., Assent, I., Baldauf, C., Seidl, T.: The clustree: indexing micro-clusters for anytime stream mining. Knowl. Inf. Syst. **29**(2), 249–272 (2011)
10. Masud, M.M., et al.: Classification and adaptive novel class detection of feature-evolving data streams. IEEE Trans. Knowl. Data Eng. **25**(7), 1484–1497 (2013)
11. Masud, M.M., Gao, J., Khan, L., Han, J., Thuraisingham, B.M.: Classification and novel class detection in concept-drifting data streams under time constraints. IEEE Trans. Knowl. Data Eng. **23**(6), 859–874 (2011)
12. Spinosa, E.J., de Leon Ferreira de Carvalho, A.C.P., Gama, J.: Novelty detection with application to data streams. Intell. Data Anal. **13**(3), 405–422 (2009)
13. Vega-Pons, S., Ruiz-Shulcloper, J.: A survey of clustering ensemble algorithms. IJPRAI **25**(3), 337–372 (2011)