Self-Adaptive Ensemble Classifier for Handling Complex Concept Drift

Imen Khamassi[1], Moamar Sayed-Mouchaweh[2]

1. Université de Tunis, Institut Supérieur de Gestion de Tunis, Tunisia
imen.khamassi@isg.rnu.tn
2. Ecole des Mines Douai, France
moamar.sayed-mouchaweh@mines-douai.fr

Abstract. In increasing number of real world applications, data are presented as streams that may evolve over time and this is known by *concept drift*. Handling concept drift through ensemble classifiers has received a great interest in last decades. The success of these ensemble methods relies on their *diversity*. Accordingly, various diversity techniques can be used like *block-based data*, *weighting-data* or *filtering-data*. Each of these diversity techniques is efficient to handle certain characteristics of drift. However, when the drift is complex, they fail to efficiently handle it. *Complex drifts* may present a mixture of several characteristics (speed, severity, influence zones in the feature space, etc) which may vary over time. In this case, drift handling is more complicated and requires new detection and updating tools. For this purpose, a new ensemble approach, namely EnsembleEDIST2, is presented. It combines the three diversity techniques in order to take benefit from their advantages and outperform their limits. Additionally, it makes use of EDIST2, as drift detection mechanism, in order to monitor the ensemble's performance and detect changes. EnsembleEDIST2 was tested through different scenarios of complex drift generated from synthetic and real datasets. This diversity combination allows EnsembleEDIST2 to outperform similar ensemble approaches in term of accuracy rate, and present stable behaviors in handling different scenarios of complex drift.
Keywords:Ensemble Classifier, Diversity techniques, Complex Concept Drift, Adaptive Learning, Evolving Data Stream, Change Detection

1 Introduction

Learning from evolving data stream has received a great attention. It addresses the non-stationarity of data over time, which is known by *concept drift*. The term *concept* refers to data distribution, represented by the joint distribution $p(x, y)$, where x represents the $n-dimensional$ feature vector and y represents its class label. The term *concept drift* refers to a change in the underlying distribution of new incoming data. For example, in intrusion detection application, the behavior of an intruder may evolve in order to confuse the system protection rules. Hence, it is essential to consider these changes for updating the system in order to preserve its performance.

Ensemble classifiers appear to be promising approaches for tracking evolving data streams. The success of the ensemble methods, according to single classifier, relies on their *diversity* [17] [22] [21]. *Diversity* can be achieved according to three main strategies [15]: *block-based data*, *weighting-data* or *filtering-data*. In *block-based ensembles* [5], [16], [20], the training set is presented as blocks or chunks of data at a time. Generally, these blocks are of equal size and the evaluation of base learners is done when all instances from a new block are available. In *weighting-data ensembles* [3] [4] [18] [13], the instances are weighted according to some weighting process. For example in Online Bagging [19], the weighting process is based on re-using instances for training individual learners. Finally, *filtering-data ensembles* [1] are based on selecting data from the training set according to a specific criterion, for example similarity in feature space.

In many real-life applications, the concept drift may be *complex* in the sense that it presents time-varying characteristics. For instance, a drift can present different characteristics according to its speed (*abrupt or gradual*), nature (*continuous or probabilistic*) and severity (*local or global*). Accordingly, *complex drift* can present a mixture of all these characteristics over time. It is worth to underline that each characteristic presents its own challenges. Accordingly, a mixture of these different characteristics may accentuate the challenge issues and complicate the drift handling.

In this paper, the goal is to underline the complementarity of the diversity techniques (*block-based data, weighting-data* and *filtering-data*) for handling different scenarios of complex drift. For this purpose, a new ensemble approach, namely EnsembleEDIST2, is proposed. The intuition is to combine these three diversity techniques in order to efficiently handle different scenarios of complex drift. Firstly, EnsembleEDIST2 defines a data-block with variable size for updating the ensemble's members, thus it can avoid the problem of tuning off size of the data-block. Secondly, it defines a new filtering criterion for selecting the most representative data of the new concept. Thirdly, it applies a new weighting process in order to create diversified ensemble's members. Finally, it makes use of EDIST2 [14] [12], as drift detection mechanism, in order to monitor the ensemble's performance and detect changes.

EnsembleEDIST2 has been tested through different scenarios of complex drifts generated from synthetic and real datasets. This diversity combination allows EnsembleEDIST2 to outperform similar ensemble approaches in term of accuracy rate, and present a stable behavior in handling different scenarios of complex drift.

The remainder of the paper is organized as follows. In *Section II*, the challenges of complex concept drift are exposed. In *Section III*, the advantages and the limits of each diversity technique are studied. In *Section IV*, the proposed approach, namely EnsembleEDIST2, is detailed. *Section V*, the experimental setup and the obtained results are presented. Finally, in *Section VI*, the conclusion and some future research directions are exposed.

2 Complex Concept Drift

In many real-life applications, the concept drift may be *complex* in the sense that it presents time-varying characteristics. Let us take the example of a drift with three different characteristics according to its speed *(gradual or abrupt)*, nature *(continuous or probabilistic)* and severity *(local or global)*. It is worth to underline that each characteristic presents its own challenges. Accordingly, a mixture of these different characteristics may accentuate the challenge issues and complicate the drift handling.

For instance, we can consider the drift depicted in Fig.1 as complex drift as it simulates a Gradual Continuous Local Drift, in the sense that the hyperplane class boundary is gradually rotating during the drifting phase and continuously presenting changes with each instance in local regions. Namely, the time until this complex drift is detected can be arbitrarily long. This is due to the rarity of data source representing the drift, which in turn makes it difficult to confirm the presence of drift. Moreover, in some cases, this drift can be considered as noise by confusion, which makes the model unstable. Hence, to overcome the instability, the model has to *(i)* effectively differentiate between local changes and noises, and *(ii)* deal with the scarcity of instances that represent the drift in order to effectively update the learner.

Another interesting complex drift represents the Gradual Continuous Global Drift (see Fig.2). During this drift, the concept is gradually changing and continuously presenting modifications with each instance. Namely, during the transition phase, the drift evolves and presents several intermediate concepts until the emergence of the final concept (see Fig.2.b). Hence, the challenging issue is to efficiently decide the end time of the old concept and detect the start time of the new concept. The objective is to update the learner with the data that represent the final concept (see Fig.2.c) and not with data collected during the concept evolution (see Fig.2.b). Moreover, this drift is considered as global because it is affecting all the instances of the drifting class. Namely, handling this complex drift is also challenging, because the performance's decrease of the learner is more pronounced than the other types of drifts.

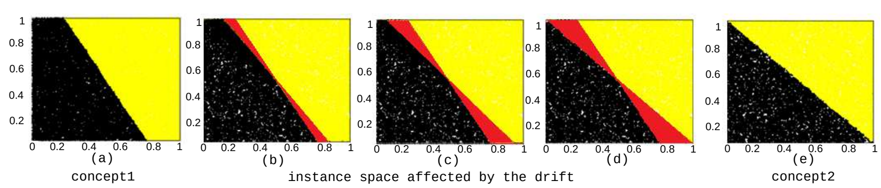

Fig. 1. Gradual Continuous Local Drift:**a** concept1, **b-d** instance space affected by the drift and **e** concept2

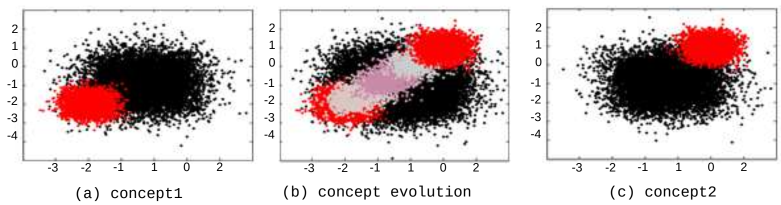

Fig. 2. Gradual Continuous Global Drift: **a** concept1, **b** concept evolution and **c** concept2

3 Related work

The *diversity* [15] among the ensemble can be fulfilled by applying various techniques such as: *block-based data*, *weighting-data* or *filtering data*, in order to differently train base learners (see Fig.3). Accordingly, the objective in this investigation is to highlight the advantages and drawbacks of each diversity techniques in handling complex drift (see Table 1).

3.1 Block-based Technique

According to the *block-based technique*, the training set is presented as blocks or chunks of data at a time. Generally, these blocks are of equal size and the construction, evaluation, or updating of base learners is done when all instances from a new block are available. Very often, ensemble learners periodically evaluate their components and substitute the weakest one with a new (candidate) learner after each data block [20] [16] [6]. This technique preserves the adaptability of the ensemble in such way that learners, which were trained in recent blocks, are the most suitable for representing the current concept.

The block-based ensembles are suitable for handling gradual drifts. Generally, during these drifts, the change between consecutive data blocks is not quite pronounced; thus, it can be only noticeable in long period. The interesting point in the block-based ensembles is that they can enclose different learners that are trained in different period of time. Hence, by aggregating the outputs of these base classifiers, the ensemble can offer accurate reactions to such gradual drifts.

In contrast, the main drawback of block-based ensembles is the difficulty of tuning off the block size to offer a compromise between fast reactions to drifts and high accuracy. If the block size is too large, they may slowly react to abrupt drift; whereas small size can damage the performance of the ensemble in stable periods.

3.2 Weighting-data Technique

In this technique, the base learners are trained according to weighted instances from the training set. A popular instance weighting process is presented in the Online Bagging ensemble [19]. For ease of understanding, the weighting process is based on re-using instances for training individual classifiers. Namely, if we consider that each base classifier C_i is trained from a subset M_i from the global training set; then the $instance_i$ will be presented k times in M_i; where the weight k is drawn from a $Poisson(1)$ distribution.

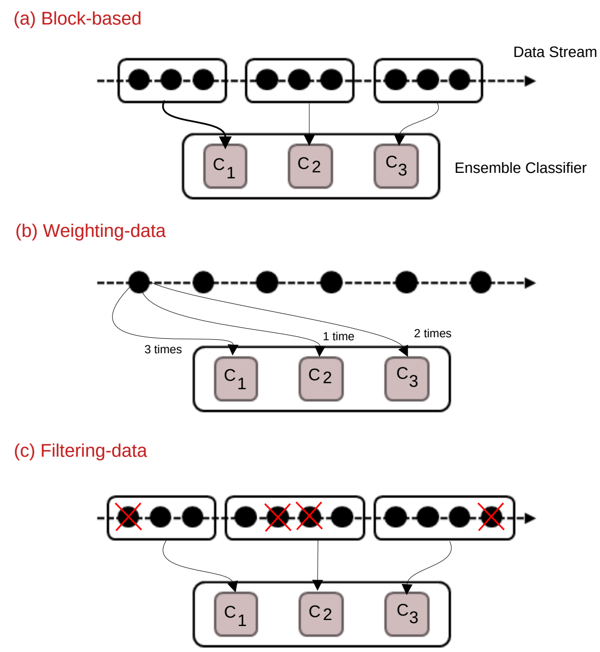

Fig. 3. Different diversity techniques among the ensemble

Online Bagging has inspired many researchers in the field of drift tracking [3] [17] [13]. This approach can be of great interest for:

- Class imbalance: where some classes are severely underrepresented in the dataset
- Local drift: where changes occur in only some regions of the instance space.

Generally, the weighting process intensifies the re-use of underrepresented class data and helps to deal with the scarcity of instances that represent the local drift. However, the instance duplication may impact the ability of the ensemble in handling global drift. During global drift, the change affects a large amount of data; thus when re-using data for constructing base classifiers, the performance's decrease is accentuated and the recovery from the drift may be delayed.

3.3 Filtering-data Technique

This technique is based on selecting data from the training set according to a specific criterion, for example similarity in the feature space. Such technique allows to select subsets of attributes that provide partitions of the training set containing maximally similar instances, i.e., instances belonging to the same regions of feature space. Thanks to this technique, base learners are trained according to different subspaces to get benefit from different characteristics of the overall feature space.

In contrast with conventional approaches which detect drift in the overall distribution without specifying which feature has changed, ensemble learners based on filtered data can exactly specify the drifting feature. This is a desired property for detecting novel class emergence or existing class fusion in unlabeled data. However, these approaches may present difficulty in handling local drifts if they do not define an efficient filtering criterion. It is worth to underline that during local drift, only some regions of the feature space are affected by the drift. Hence, only the base classifier which is trained on changing region is the most accurate to handle the drift. However, when aggregating the final decision of this classifier with the remained classifiers, trained from unchanged regions, the performance recovery may be delayed.

Table 1. Summary of the advantages (+) and drawbacks (-) of diversity techniques for handling complex drift

Complex Drift	Gradual Continuous		Gradual Probabilistic		Abrupt	
	Local	Global	Local	Global	Local	Global
Block-based	+	+	+	+	-	-
Weighting-data	+	-	+	-	+	-
Filtering-data	-	+	-	+	-	+

4 The proposed approach

The intuition behind EnsembleEDIST2 is to combine the three diversity techniques (*Block-based*, *Weighting-data* and *Filtering data*) in order to take benefit from their advantages and avoid their drawbacks.

The contributions of EnsembleEDIST2 for efficiently handling complex concept drifts are as follows, it:

- Explicitly handles drift through a drift detection method EDIST2 [14] (subSection4.1)
- Makes use of data-block with variable size for updating the ensemble's members (subSection4.2)

– Defines a new filtering criterion for selecting the most representative data of the new concept (subSection4.3)
– Applies a new weighting process in order to create diversified ensemble's members (subSection4.4)

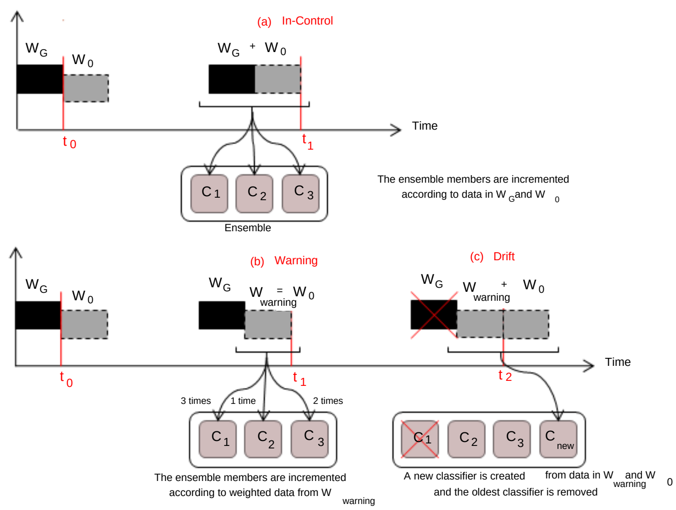

Fig. 4. EnsembleEDIST2's adapting process according to the three detection levels: (a) In-control, (b) Warning and (c) Drift

4.1 Drift monitoring process in EnsembleEDIST2

EnsembleEDIST2 is an ensemble classifier designed to explicitly handle drifts. It makes use of EDIST2 [14], as drift detection mechanism, in order to monitor the ensemble's performance and detect changes (see Fig4).

EDIST2 monitors the prediction feedback provided by the ensemble. More precisely, EDIST2 studies the distance between two consecutive errors of classification. Notice that the distance is represented by the number of instances between two consecutive errors of classification. Accordingly, when the data distribution becomes non-stationary, the ensemble will commit much more errors and the distance between these errors will decrease.

In EDIST2, the concept drift is tracked through two data windows, a 'global' one and a 'current' one. The global window W_G is a self-adaptive window which is continuously incremented if no drift occurs and decremented otherwise; and the current window W_0 which represents the batch of current collected instances.

In EDIST2, we want to estimate the error distance distribution of W_G and W_0 and make a comparison between the averages of their error distance distributions in order to check a difference. As stated before, a significant decrease in the error distance implies a change in the data distribution and suggests that the learning model is no longer appropriate.

EDIST2 makes use of a statistical hypothesis test in order to compare W_G and W_0 error distance distributions and check whether the averages differ by more than the threshold ϵ. It is worth underlining that there is no *a priori* definition of the threshold ϵ, in the sense that it does not require any *a priori* adjusting related to the expected speed or severity of the change. ϵ is autonomously adapted according to a statistical hypothesis test (for more details please refer ti [14]).

The intuition behind EDIST2 is to monitor μ_d which represents difference between W_G and W_0 averages and accordingly three thresholds are defined:

- *In-Control level*: $\mu_d \leq \epsilon$; within this level, we confirm that there is no change between the two distributions, so we enlarge W_G by adding W_0 's instances. Accordingly, all the ensemble members are incremented according to data samples in W_G and W_0.
- *Warning level*: $\mu_d > \epsilon$; within this level, the instances are stored in an warning chunk $W_{warning}$. Accordingly, all the ensemble members are incremented according to weighted data from $W_{warning}$. (The weighting process will be explained in *subSection4.4*)
- *Drift level*: $\mu_d > \epsilon + \sigma_d$; within this level, the drift is confirmed and W_G is decremented by only containing the instances stored since the warning level,i.e., in $W_{warning}$. Additionally, a new base classifier is created from scratch and trained according to data samples in $W_{warning}$, then the oldest classifier is removed from the ensemble.

4.2 EnsembleEDIST2's diversity by variable-sized block technique

In EnsembleEDIST2, the size of data-block is not defined according to the number of instances, as it is the case of conventional *block-based ensembles*, but according to the number of errors committed during the learning process. More precisely, the data-block W_0, in EnsembleEDIST2, is constructed by collecting the instances that exist between N_0 errors.

As depicted in Fig.5 , when the drift is abrupt, the ensemble commits N_0 errors in short drifting time. However, when the drift is gradual, the ensemble commits N_0 errors in relatively longer drifting time. Hence, according to this strategy, the block size is variable and adjusted according to drift characteristics.

It is worth to underline that EnsembleEDIST2 can offer a compromise between fast reaction to abrupt drift and stable behavior regarding gradual drift. This is a desirable property for handling complex drift which may present different characteristics in the same time, and accordingly EnsembleEDIST2 can avoid the problem of tuning off the size of data-block as it is the case of most block-based approaches.

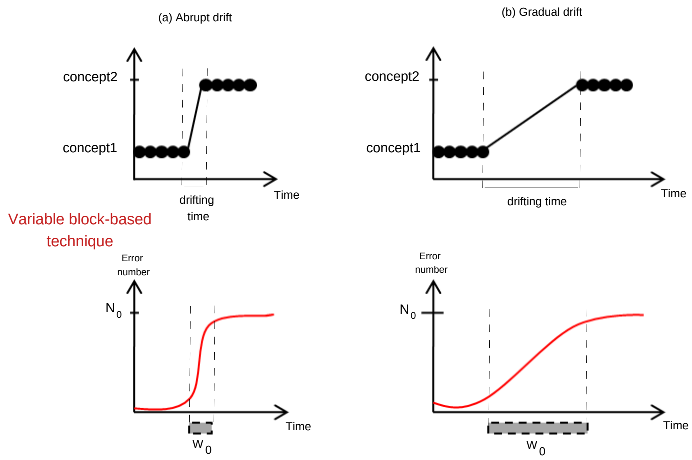

Fig. 5. Variable data-block technique in EnsembleEDIST2

4.3 EnsembleEDIST2's diversity by new filtering-data criterion

Differently from conventional filtering-data ensembles, which filter data according to similarity in the feature space, EnsembleEDIST2 defines a new filtering criterion. It filters the instances that trigger the warning level. More precisely, each time the ensemble reaches the warning level, the instances are gathered in a warning chunk $W_{warning}$ in order to re-use them for training the ensemble's members (see Fig.6.a). This is an interesting point when dealing with local drift because drifting data are scarce and not continuously provided. It is possible that a certain amount of drifting data can be found in zones (1), (2), (3) and (4) but not quite sufficient to reach the drift level. Accordingly, by considering these data for updating the ensemble's members, EnsembleEDIST2 can ensure a rapid recovery from local drift.

In contrast, conventional filtering-data ensembles are unable the define in which zone the drift has occurred, thus, they may update the ensemble's members with data filtered from unchanged feature space; which in turn may delay the performance correctness.

4.4 EnsembleEDIST2's diversity by new weighting-data process

The focus in EnsembleEDIST2 is to maximize the use of data present in $W_{warning}$ for accurately updating the ensemble. More precisely, the data in $W_{warning}$ are weighted according to the same weighting process used in Online bagging [19]. Namely, each $instance_i$ from $W_{warning}$ is re-used k times for training the base classifier C_i , where the weight k is drawn from a $Poisson(1)$ distribution (see Appendix7).

Generally, the weighting process in EnsembleEDIST2 offers twofold advantages. First, it intensifies the re-use of underrepresented class data and helps to deal with scarcity of instances that represent the local drift. Second, it permits faster recovery from global drift than conventional weighting-data ensembles. As it is known, during global drift, the change affects a large amount of data. Hence, differently from conventional weighting-data ensembles, which apply the weighting process to all the data sets; EnsembleEDIST2 only weights the instances present in $W_{warning}$ (see Fig.6.b). Accordingly, it can avoid to accentuate the decrease of the ensemble's performance during global drift, and ensure a fast recovery.

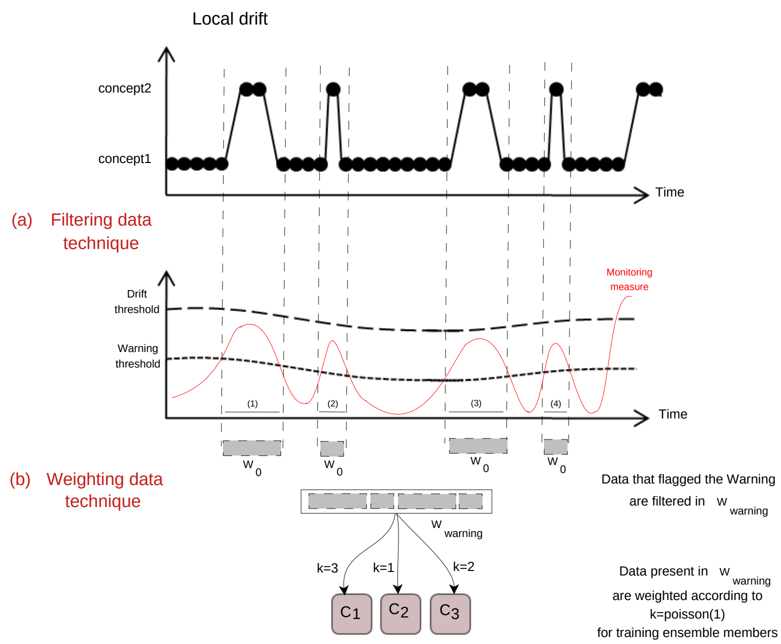

Fig. 6. (a) Filtering-data technique and (b) Weighting-data technique in EnsembleEDIST2

5 Experiments and performance analysis

5.1 Experimental evaluation

Synthetic Datasets In this investigation, we are studying six different scenarios of complex concept drift as depicted in Table 2 . All synthetic datasets contain $100,000$ instances and one concept drift where the starting and the ending time are predefined. For gradual drift, the drifting time lasts $30,000$ instances (it begins at t_{start}=40,000 and ends at $t_{end} = 70,000$). For abrupt drift, the drift occurs at $t = 50,000$.

Table 2. Different types of Complex Drift handled in this investigation

Complex Drift Characteristics			Synthetic Datasets
Speed	**Nature**	**Severity**	
Gradual	Continuous	Local	Hyperplane [10]
		Global	RBF [2]
	Probabilistic	Local	SEA Gradual [24]
		Global	STAGGER Gradual [23]
Abrupt		Local	SEA Abrupt [24]
		Global	STAGGER Abrupt [23]

Real Datasets

Electricity Dataset (48,312 instances, 8 attributes, 2 classes) is a real world dataset from the Australian New South Wales Electricity Market [9]. In this electricity market, the prices are not fixed and may be affected by demand and supply. The dataset covers a period of two years and the instances are recorded every half an hour. The classification task is to predict a rise (UP) or a fall (DOWN) in the electricity price. Three numerical features are used to define the feature space: the electricity demand in current region, the electricity demand in the adjacent regions and the schedule of electricity transfer between the two regions.

This dataset may present several scenarios of complex drift. For instance, a gradual continuous drift may occur when the users progressively change their consumption habits during a long time period. Likewise, an abrupt drift may occur when the electricity prices suddenly increase due to unexpected events (e.g., political crises or natural disasters). Moreover, the drift can be local if it impacts only one feature (e.g., the electricity demand in current region); or global if it impacts all the features.

Spam Dataset (9,324 instances, 500 attributes, 2 classes) is a real world dataset containing email messages from the Spam Assassin Collection Project [11]. The classification task is to predict if a mail is a spam or legitimate. The data set contains 20% of spam mailing. The feature space is defined by a set of numerical features such as the number of receptors, textual attributes describing the mail contain and sender characteristics...

This dataset may present several scenarios of complex drift. For instance, a gradual drift may occur when the user progressively changes his preferences. However, an abrupt drift may occur when the spammer rapidly changes the mail content to trick the spam filter rules. It is worth to underline that the drift can also be continuous when the spammer starts to change the spam content; but the filter continues to correctly detect them. In the other side, the drift can be probabilistic when the spammer starts to change the spam content; but the filter fails in detecting some of them.

Evaluation criteria When dealing with evolving data streams, the objective is to study the evolution of the EnsembleEDIST2 performance over time and see how quick the adaptation to drift is. According to Gama et al. [8] the prequential accuracy is a suitable metric to evaluate the learner performance in presence of concept drift. It proceeds as follows: each instance is firstly used for testing then for training. Hence, the accuracy is incrementally updated using the maximum available data; and the model is continuously tested on instances that it has not already seen (for more details please refer to [8]).

Parameter Settings All the tested approaches were implemented in the java programming language by extending the Massive Online Analysis (MOA) software [2]. MOA is an online learning framework for evolving data streams and supports a collection of machine learning methods.

For comparison, we have selected well known ensemble approaches according to each category:

- *Block-based ensemble*: AUE (Accuracy Updated Ensemble) [5], AWE (Accuracy Weighted Ensemble) [16] and LearnNSE [20] with block size equal to 500 instances.
- *Weighting-data ensemble*: LeveragingBag [3] and OzaBag [19]
- *Filtering-data ensemble*: LimAttClass [1]

For all these approaches, the ensemble's size was fixed to 10 and the Hoeffding Tree (HT) [7] was used as base learning algorithm.

It is worth to notice that EnsembleEDIST2 makes use of two parameters: N_0 which is the number of error in W_0 and m which is the number of base classifiers among the ensemble. In this investigation, we respectively set $N_0 = 30$ and $m = 3$ according to empirically studies done in *subSections* 5.2 and 5.2.

5.2 Comparative study and interpretation

Impact of N_0 on EnsembleEDIST2 performance EnsembleEDIST2 makes use of the parameter N_0 in order to define the minimum number of error occurred in W_0. Recall that W_0 represents the batch of current collected instances. This batch is constructed by collecting the instances that exist between N_0 errors.

It is interesting to study the impact of N_0 on the accuracy according to different scenarios of complex drift. For this purpose, we have done the following experiments: for each scenario of complex drift, the accuracy of EnsembleEDIST2 is presented by varying N_0 values (see Table 3).

Based on these results, we can conclude that the performance of EnsembleEDIST2 in handling different scenarios of complex drifts is weakly sensitive to N_0. Hence, we have decided to use $N_0 = 30$ as it has achieved the best accuracy rate in most cases.

Table 3. Prequential accuracy for different values of N_0 in EnsembleEDIST2

Complex drift	Gradual Continuous		Gradual Probabilistic		Abrupt	
	Local	Global	Local	Global	Local	Global
Synthetic database	Hyperplane	RBF	SEA Gradual	STAGGER Gradual	SEA Abrupt	STAGGER Abrupt
$N_0 = 30$	**98,6**	**95,9**	**97,2**	91,6	97,9	**99,6**
$N_0 = 60$	98,2	**95,9**	**97,2**	91,5	98,1	**99,6**
$N_0 = 90$	98,2	95,6	97,1	91,6	97,5	**99,6**
$N_0 = 120$	98,3	**95,9**	97,1	91,6	**98,2**	**99,6**
$N_0 = 150$	98,3	95,6	97,1	**91,7**	97,5	**99,6**

Impact of ensemble size on EnsembleEDIST2 performance EnsembleEDIST2 makes use of the parameter m in order to define the number of classifiers in the ensemble. Accordingly, it is interesting to study the impact of m on ensemble's performance according to different scenarios of complex drift.

According to Table4, it is noticeable that the size of EnsembleEDIST2 does not impact significantly the performance in handling different scenarios of complex drift. Hence, we have decided to use $m = 3$ as it achieved the best accuracy rate in most cases and it allows to limit the computational complexity of the ensemble.

Table 4. Accuracy of EnsembleEDIST2 with different number of base classifiers

Complex drift	Gradual Continuous		Gradual Probabilistic		Abrupt	
	Local	Global	Local	Global	Local	Global
Synthetic database	Hyperplane	RBF	SEA Gradual	STAGGER Gradual	SEA Abrupt	STAGGER Abrupt
$m = 3$	**98,6**	**95,9**	**97,2**	91,6	97,9	**99,6**
$m = 5$	**98,6**	**95,9**	97,1	91,6	**98**	**99,6**
$m = 10$	98.4	95,8	**97,2**	**91,1**	97,6	**99,6**

Accuracy of EnsembleEDIST2 Vs other ensembles Table5 summarizes the average of prequential accuracy during the drifting phase. The objective of this experiment is to study the ensemble performance in the presence of different scenarios of complex drift. Firstly, it is noticeable that EnsembleEDIST2 has achieved better results than *block-based ensembles* in handling different types of abrupt drift. During abrupt drift (independently of being local of global), the change is rapid; thus AUE, AWE and LearnNSE present difficulty in tuning off the block size to offer a compromise between fast reaction to drift and high accuracy. However, EnsembleEDIST2 is able to autonomously train ensemble members with variable amount of data at each time process, thus it can efficiently handle abrupt drift.

Secondly, it is noticeable that EnsembleEDIST2 outperforms *weighting-data ensembles* in handling different categories of global drift. During global drift (either continuous, probabilistic or abrupt), the change affects a large amount of data; thus when LeveragingBag and OzaBag intensify the re-use of data for training ensemble members, the performance's decrease is accentuated. In contrast, EnsembleEDIST2 duplicates only a set of filtered instances for training the ensemble members, that is why it is more accurate in handling global drift.

Thirdly, it is noticeable that EnsembleEDIST2 outperforms the *filtering-data ensembles* in handling different categories of local drift. During local drift (either continuous, probabilistic or abrupt), the change affects a little amount of data; thus the choice of the filtering criterion is a essential point for efficiently handling local drift. EnsembleEDIST2 defines a new filtering criterion, which is based on selecting the data that triggered the warning level. These data are the most representative of the new concept, thus when training the ensemble's members accordingly, it makes it more efficient for handling local drift.

EnsembleEDIST2 has also been tested through real world data sets which represent different scenarios of drift. It is worth underlining that the size of these data sets is relatively small comparing to the synthetic ones. Despite the different features of each real data set, encouraging results have been found where EnsembleEDIST2 has achieved the best accuracy in all the datasets (see Table6).

To sum, it is worth to underline that the combination of the three diversity techniques in EnsembleEDIST2 is beneficial for handling different scenarios of complex drift in the same time.

Table 5. Accuracy of EnsembleEDIST2 Vs. other ensembles in synthetic datasets

Complex Drift		Gradual Continuous		Gradual Probabilistic		Abrupt	
		Local	Global	Local	Global	Local	Global
Synthetic Dataset		Hyperplane	RBF	SEA Gradual	STAGGER Gradual	SEA Abrupt	STAGGER Abrupt
EnsembleEDIST2		**98,604**	**95,982**	**97,211**	**91,609**	**98,196**	**99,605**
Block-based	AUE	94,187	95,611	94,547	90,381	95,234	98,367
	AWE	94,054	95,018	94,563	90,551	95,23	98,367
	LearnNSE	96,369	95,44	94,372	85,873	95,079	39,049
Weighting-data	LeveragingBag	98,6	95,8	97,1	89,1	**98,2**	94,3
	OzaBag	98,195	93,533	96,982	69,21	98,132	96,64
Filtering-data	LimAttClass	91,281	94,186	91,126	86,553	91,226	94,893

Table 6. Accuracy of EnsembleEDIST2 Vs. other ensembles in real datasets

Real Dataset		Electricity	Spam
EnsembleEDIST2		**84,8**	**89,2**
Block-based	AUE	69,35	79,34
	AWE	72,09	60,25
	LearnNSE	72,07	60,33
Weighting-data	LeveragingBag	83.8	88,2
	OzaBag	82,3	82,7
Filtering-data	LimAttClass	82,6	63,9

6 Conclusion

In this paper, we have presented a new study of the role of diversity among the ensemble. More precisely, we have highlighted the advantages and the limits of three widely used diversity techniques (*block-based data*, *weighting-data* and *filtering data*) in handling *complex drift*.

Additionally, we have presented a new ensemble approach, namely EnsembleEDIST2, which combines these three diversity techniques. The intuition behind this approach is to explicitly handle drifts by using the drift detection mechanism EDIST2. Accordingly, the ensemble performance is monitored through a self-adaptive window. Hence, EnsembleEDIST2 can avoid the problem of tuning off the size of the batch data as it is the case of most block-based ensemble approaches, which is a desirable property for handling abrupt drifts. Secondly, it defines a new filtering criterion, which is based on selecting the data that trigger the warning level. Thanks to this property, EnsembleEDIST2 is more efficient for handling local drifts then conventional filtering-data ensembles, which are only based on filtering data according to similarity on feature space. Then, differently from the conventional weighting-data ensembles which apply the weighting process to all the data stream; EnsembleEDIST2 only intensifies the re-use of most representative data of the new concept, which is a desirable property for handling global drifts.

EnsembleEDIST2 has been tested different scenarios of complex drift. Encouraging results were found, comparing to similar approaches, where EnsembleEDIST2 has achieved the best accuracy rate in all datasets; and presented a stable behavior in handling different scenarios of complex drift.

It worth to underline that in the present investigation, the ensemble size, i.e., the number of ensemble members, was fixed. Hence it is interesting, for future work, to perform a strategy for dynamically adapting the ensemble size. The focus is that, during stable period, the ensemble size is maintained fixed; whereas during the drifting phase the size is autonomously adapted. This may ameliorate the performance and reduce the computational cost among the ensemble.

Acknowledgements The second author acknowledges the support of the Regional project REPAR, funded by the French Region Hauts-de-France.

7 EnsembleEDIST2 pseudo code

Algorithm *EnsembleEDIST2*
Input: (x, y): Data Stream
  N_0: number of error to construct the window
  m: number of base classifier
Output: Trained ensemble classifier E

for each base classifier C_i from E
  $InitializeClassifier(C_i)$
end for
$W_G \leftarrow CollectInstances(E, N_0)$
$W_{warning} \leftarrow \emptyset$
repeat
  $W_0 \leftarrow CollectInstances(E, N_0)$
  $Level \leftarrow DetectedLevel(W_G, W_0)$
  switch (Level)
  case 1: $Incontrol$
    $W_G \leftarrow W_G \cup W_0$
    $UpdateParameters(W_G, W_0)$
    Increment all ensemble's members of E according to instances in W_G
  end case 1
  case 2: $Warning$
    $W_{warning} \leftarrow W_{warning} \cup W_0$
    $UpdateParameters(W_{warning}, W_0)$
    $WeightingDataProcess(E, W_{warning})$
  end case 2
  case 3: $Drift$
    Create a new base classifier C_{new} trained on instances in $W_{warning}$
    $E \leftarrow E \cup C_{new}$
    Remove the oldest classifier from E
    $W_G \leftarrow W_{warning}$
    $W_{warning} \leftarrow \emptyset$
  end case 3
end switch
until The end of the data streams

Algorithm *DetectedLevel*(W_G, W_0)
Input: W_G: Global data window characterized by:
    N_G: error number
    μ_G: error distance mean
    σ_G:error distance standard deviation
  W_0: Current data window characterized by:
    N_0: error number,

μ_0: error distance mean,
σ_0:error distance standard deviation
Output: $Level$: detection level
$\mu_d \leftarrow \mu_G$-μ_0
$\sigma_d \leftarrow \sqrt{\frac{\sigma_G^2}{N_G} + \frac{\sigma_0^2}{N_0}}$
$\epsilon \leftarrow t_{1-\alpha} * \sigma_d$
if $(\mu_d > \epsilon + \sigma_d)$
    $Level \leftarrow Drift$
    else if $(\mu_d > \epsilon)$
        $Level \leftarrow Warning$
        else $Level \leftarrow Incontrol$
    end if
end if
return (Level)

Algorithm *CollectInstances(E, N_0)*
Input: (x, y): Data Stream
N_0: number of error to construct the window
C: trained ensemble classifier E
Output: W: Data window characterized by:
N: error number
μ: error distance mean
σ:error distance standard deviation
$W \leftarrow \emptyset$
$N \leftarrow 0$
$\mu \leftarrow 0$
$\sigma \leftarrow 0$
repeat for each instance x_i
    $Prediction \leftarrow unweightedMajorityVote(E, x_i)$
    if $(Prediction = false)$
        $d_i \leftarrow computeDistance()$
        $\mu \leftarrow \frac{N}{N+1}\mu + \frac{d_i}{N+1}$
        $\sigma \leftarrow \sqrt{\frac{N-1}{N}\sigma^2 + \frac{(d_i-\mu)^2}{N+1}}$
        $N \leftarrow N + 1$
    end if
    $W \leftarrow W \cup \{x_i\}$
until $(N = N_0)$
return (W)

Algorithm *UpdateParameters(W_G, W_0)*
Input: W_G: Global data window characterized by:
N_G: error number

μ_G: error distance mean
σ_G:error distance standard deviation
W_0: Current data window characterized by:
N_0: error number,
μ_0: error distance mean,
σ_0:error distance standard deviation
Output: Updated parameters of W_G

$\mu_G \leftarrow \frac{1}{N_G+N_0}(N_G.\mu_G+N_0.\mu_0)$ $\sigma_G \leftarrow \sqrt{\frac{N_G\sigma_G^2+N_0\sigma_0^2}{N_G+N_0} + \frac{N_G N_0}{(N_G+N_0)^2}(\mu_G-\mu_0)^2}$
$N_G \leftarrow N_G + N_0$

Algorithm *WeightingDataProcess*$(E, W_{warning})$
Input: E: Ensemble Classifier
$W_{warning}$: Window of data
Output: E: Updated ensemble classifier

for each instance x_i from $W_{warning}$
    for each base classifier C_i from E
        $k \leftarrow poisson(1)$
        do k times
            $TrainClassifier(C_i, x_i)$
        end do
    end for
end for

References

1. Bifet, A., Frank, E., Holmes, G., Pfahringer, B., Sugiyama, M., Yang, Q.: Accurate ensembles for data streams: Combining restricted hoeffding trees using stacking. In: 2nd Asian Conference on Machine Learning (ACML2010). pp. 225–240 (2010)
2. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: massive online analysis. Journal of Machine Learning Research 11, 1601–1604 (2010)
3. Bifet, A., Holmes, G., Pfahringer, B.: Leveraging bagging for evolving data streams. In: Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part I. pp. 135–150. ECML PKDD'10, Springer-Verlag, Berlin, Heidelberg (2010), `http://dl.acm.org/citation.cfm?id=1888258.1888275`
4. Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., Gavaldà, R.: New ensemble methods for evolving data streams. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 139–148. KDD '09, ACM, New York, NY, USA (2009), `http://doi.acm.org/10.1145/1557019.1557041`
5. Brzezinski, D., Stefanowski, J.: Reacting to different types of concept drift: The accuracy updated ensemble algorithm. Neural Networks and Learning Systems, IEEE Transactions on 25(1), 81–94 (Jan 2014)

6. Brzezinski, D., Stefanowski, J.: Accuracy updated ensemble for data streams with concept drift. In: Corchado, E., Kurzy?ski, M., Wo?niak, M. (eds.) Hybrid Artificial Intelligent Systems, Lecture Notes in Computer Science, vol. 6679, pp. 155–163. Springer Berlin Heidelberg (2011), `http://dx.doi.org/10.1007/978-3-642-21222-2_19`
7. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 71–80. KDD00, ACM, New York, NY, USA (2000)
8. Gama, J., Sebastio, R., Rodrigues, P.: On evaluating stream learning algorithms. Machine Learning 90(3), 317–346 (2013)
9. Harries, M.: Splice-2 comparative evaluation: Electricity pricing. Tech. rep., The University of South Wales, United Kingdom (1999)
10. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001. pp. 97–106 (2001)
11. Katakis, I., Tsoumakas, G., Vlahavas, I.: Tracking recurring contexts using ensemble classifiers: an application to email filtering. Knowledge and Information Systems 22(3), 371–391 (2010)
12. Khamassi, I., Sayed-Mouchaweh, M.: Drift detection and monitoring in non-stationary environments. In: Evolving and Adaptive Intelligent Systems (EAIS), Austria. pp. 1–6 (June 2014)
13. Khamassi, I., Sayed-Mouchaweh, M., Hammami, M., Ghédira, K.: Ensemble classifiers for drift detection and monitoring in dynamical environments. In: Annual Conference of the Prognostics and Health Management Society, New Orlean, USA, 2013 (October 2013)
14. Khamassi, I., Sayed-Mouchaweh, M., Hammami, M., Ghédira, K.: Self-adaptive windowing approach for handling complex concept drift. Cognitive Computation 7(6), 772–790 (2015), `http://dx.doi.org/10.1007/s12559-015-9341-0`
15. Khamassi, I., Sayed-Mouchaweh, M., Hammami, M., Ghédira, K.: Discussion and review on evolving data streams and concept drift adapting. Evolving Systems (Oct 2016), `http://dx.doi.org/10.1007/s12530-016-9168-2`
16. Kolter, J.Z., Maloof, M.A.: Dynamic weighted majority: An ensemble method for drifting concepts. J. Mach. Learn. Res. 8, 2755–2790 (Dec 2007)
17. Minku, L., White, A., Yao, X.: The impact of diversity on online ensemble learning in the presence of concept drift. Knowledge and Data Engineering, IEEE Transactions on 22(5), 730–742 (May 2010)
18. Minku, L., Yao, X.: Ddd: A new ensemble approach for dealing with concept drift. Knowledge and Data Engineering, IEEE Transactions on 24(4), 619–633 (April 2012)
19. Oza, N.C., Russell, S.: Online bagging and boosting. In: In Artificial Intelligence and Statistics 2001. pp. 105–112. Morgan Kaufmann (2001)
20. Polikar, R., Upda, L., Upda, S., Honavar, V.: Learn++: an incremental learning algorithm for supervised neural networks. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 31(4), 497–508 (Nov 2001)
21. Ren, Y., Zhang, L., Suganthan, P.N.: Ensemble classification and regression-recent developments, applications and future directions [review article]. IEEE Computational Intelligence Magazine 11(1), 41–53 (2016)
22. Sayed-Mouchaweh, M.: Learning from Data Streams in Dynamic Environments, chap. Handling Concept Drift, pp. 33–59. Springer International Publishing, Cham (2016), `http://dx.doi.org/10.1007/978-3-319-25667-2_3`

23. Schlimmer, J.C., Granger, Jr., R.H.: Incremental learning from noisy data. Mach. Learn. 1(3), 317–354 (Mar 1986)
24. Street, W.N., Kim, Y.: A streaming ensemble algorithm (sea) for large-scale classification. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 377–382. KDD01, ACM, New York, NY, USA (2001)