

Analyzing Adaptive and Non-Adaptive Online Learners on Imbalanced Evolving Streams

Himaja. D¹, Dondeti Venkatesulu², Uppalapati Srilakshmi³

¹ Computer Science and Engineering, Vignan's Foundation for Science and Technology (Deemed to be University), Guntur, Andhra Pradesh, India

^{2,3} Advanced Computer Science and Engineering, Vignan's Foundation for Science and Technology (Deemed to be University), Guntur, Andhra Pradesh, India

E-mail: himajadirsumilli@gmail.com

Keywords: online learners, dynamic class imbalance, concept drift, active learning, support vector machine

Received: November 22, 2022

The online class imbalance and concept drift (OCI-CD) has recently received much interest. The impact of this combined problem on state-of-the-art of online adaptive and non-adaptive learners has received little attention. This study investigates the effect of parameters such as current imbalance ratio, stream length, drift type, drift levels, and imbalance state (static or dynamic) on adaptive and non-adaptive online learners. The experimental results show that each parameter considered for the study has a significant impact on learner performance: (a) minority class performance decreases as the degree of imbalance increases, (b) non-adaptive learners are much susceptible to class imbalance, concept drift, and the combined problem of both drifts than adaptive learners, (c) adaptive learners are only susceptible to class imbalance drifts, and (d) the impact of the dynamic degree of imbalance is more on learner than static (e) the adaptive large scale support vector machine yields stable performance to all the parameters considered for the study. Based on these findings, directions for developing new approaches are also presented.

Povzetek: Analizirane so razne metode strojnega učenja glede na parametre učenja, recimo spreminjanje neuravnoteženja razredov.

1 Introduction

Real-world classification problems like fraud and fault detection are constantly changing due to class imbalance (CI) and concept drift (CD) [1, 2, 3]. One class of samples in a stream will experience CI if it is much smaller [4, 5]. The CI between classes changes with time in evolving streams (i.e., dynamic) [3]. When the underlying function that generates concepts changes, CD happens.

Let the input (x) and goal (y) variables be included in the training dataset. The Bayesian theorem states that three different types of drifts can result from changes in (i) the posterior $p(y/x)$, (ii) the prior $p(y)$, without changing $p(y/x)$ and $p(x/y)$, and (iii) the likelihood $p(x/y)$, without affecting $p(y/x)$ and $p(y)$ [3]. Real CDs are changes in (i) over time that don't depend on changes in $p(y)$ and $p(x/y)$. The sorts of drifts (ii) and (iii) on the other hand are virtual CDs [1, 2, 3]. Real and virtual CDs coexist in the real world. There are three types of drifts based on the speed of evolution: (i) gradual, where the concept changes gradually (ii) abrupt, where the underlined concept changes suddenly and (iii) recurrent (or cyclic), where the same concept recurs regularly [1]. Thus, an imbalanced stream evolving with CD is an online class imbalance with CD (OCI-CD) problem.

An adaptive learner is a window-based technique that preserves the training samples from the current time t while ignoring the older samples and only using representative samples from the window [1, 2]. A non-

adaptive learner, on the other hand, does not employ a window for incremental learning streams. The typical application of non-adaptive learning strategies is in static domains [1, 2, 3].

2 Motivation

Wang et al. [3] recently conducted a systematic analysis to determine the impact of CI on three different types of drifts (i.e., $p(y)$, $p(y/x)$, and $p(x/y)$) while ignoring their counterparts. The effect of different levels of imbalance (static and dynamic) and coupled dynamic OCI-CD drift on online learner's performance has yet to be empirically investigated. They only gave advice based on observations of cases with a high level of imbalance (i.e., 1:9).

3 Contributions

To fill in the gaps mentioned above, the impact of the training stream's characteristics, including the degree of imbalance, length at the time t , drift types (CI, CD, and OCI-CD), and the state of imbalance (static and dynamic) on state-of-the-art adaptive and non-adaptive learners used for minority class prediction, is explored.

This study explores various static and dynamic imbalanced streams with gradual and abrupt drift levels. This work also aims to answer the following research questions:

RQ1. Does the length of the stream with respect to the imbalance ratio at the current time t impact the online learner's performance?

RQ2. Is the degree of imbalance or CD whose impact is critical on minority class performance degradation?

RQ3. Is the impact of OCI-CD more adverse than individual $p(y)$ or $p(y/x)$ drifts on the learner's performance?

RQ4. To what extent does online SVM cope with OCI-CD, $p(y)$, and $p(y/x)$ drifts compared to other online learners?

The case of combined $p(x/y)$, $p(y/x)$ drift is not considered in the scope of this study, as the impact is only $p(x/y)$ due to the change in the likelihood of the concept [3].

The following is the structure of the paper. The related work is shown in Section 4. Section 5 provides background for the problem-related methods. Section 6 discusses the study design, while section 7 discusses the experiments performed on the synthetic data. Section 8 discusses the validity of the observations on real-world data, and section 9 discusses the results obtained in greater depth. Section 10 brings this paper to a close.

4 Related work

4.1 CI problem

This problem has solutions at both the algorithmic and data levels [4, 5]. Solutions include resampling techniques at the data level. Adjusting the threshold [6], cost-sensitive learning [7, 8], and novelty detection techniques [9] are examples of algorithm-level solutions.

Ensemble learning techniques like bagging and boosting has been intensively studied as solutions to the CI problem. Cost-sensitive learning-based boosting [10, 11, 12], under-over bagging [13, 14], under-sampling-based boosting [15], oversampling-based boosting [16, 17], under sampling-based bagging [18, 19], oversampling based bagging [13], and hybrids of bagging and boosting, under-over-bagging [13, 14] are proposed hybrid ensembles that improve minority class prediction.

4.2 Learning streams from non-stationary environments

Real and virtual drifts, or a combination of the two, can be found in an evolving stream. On the topic of drift detection, there has been a lot of research, including recent surveys [1, 2, 20]. These categorize drift detection techniques into two categories: (i) active and (ii) passive.

The active methods detect drift first and then update/rebuild the learner to adapt to data changes. The drift detection can be carried out by hypothesis tests [21, 22], change-point method [23], sequential hypothesis test [24], and change detection test [23]. Recently, statistical methods that identify distribution differences have been used in SDDM [25] for drift detection, while cluster-based distance methods [26] has been used to detect recurring CDs. Drift detection methods such as the drift detection method for OCI (DDM-OCI) [27], LFR [28], and PAUC

[29], on the other hand, detect $p(y/x)$ drift in imbalanced distributions. Wang et al. defined AUC for multi-class classification as prequential multiclass AUC (PMAUC), weighted AUC (WAUC), and equal-weighted AUC (EWAUC) [30].

When it comes to passive learning techniques, in contrast to detection and adaptation methods, the highlighted model continuously adapts to the change by updating a single classifier or adding/removing/modifying a classifier in an ensemble [1] to retain the new knowledge and forget the old for each new set of data. A heterogeneous dynamic weighted majority (HDWM) [31] is suggested to replace the existing base learner in the ensemble when a performance decrease is seen. It works with both active and passive strategies. Even though they require more computing, ensemble approaches are superior to single learners. While passive approaches are better for gradual drifts, active ones are better for batch learning and forecasting rapid drifts [1].

4.3 Learning imbalanced streams from non-stationary environments

This is the issue that arises when CD and CI streams coevolve. Either a static or dynamic evolution of the CI stream is possible. The term "dynamic" in this context refers to the $p(y)$ change [3], or the dynamic change in CI degree.

Gao et al. [32] proposed an instance propagation ensemble mechanism. Chen and he [33] proposed selecting the best n minority class instances using mahalanobis distance. Lichten and Chawla [34, 35, 36] proposed an extension to the work of gao et al. [32]. Instead of simply propagating minority class samples, they proposed method that propagates misclassified majority class instances from the previous model. They also proposed weighing every ensemble member based on the probability of a combined hellinger distance and information gain [34] change detection test. HUWRS.IP [36], an instance selection method based on the NB classifier is proposed.

Learn++CDS and Learn++ are two ensemble approaches proposed as an extension to Learn++.NSE [37], to handle the OCI-CD problem from the evolving stream. The former employs the SMOTE oversampling technique to rebalance the data, while the latter employs the bagging-based sub-ensemble method. Wang et al. [38] proposed a resampling-based ensemble method (OOB/UOB) for online bagging in which the time decay class size guides oversampling and under sampling rates.

Active approaches with single classifiers include recursive least square adaptive cost perceptron (RLSACP) [39] and online neural network (ONN) [40]. An ensemble of the subset of online sequential extreme learning machine (ESOS-ELM) [41] is proposed. Baruva et al. [42] proposed a generalized over-sampling-based online imbalanced learning framework (GOS-IL) for online learners to only cope with $p(y)$ drift.

Lu et al. [43] proposed dynamic weighted majority for imbalance learning (DWMIL), a batch based incremental learning method to deal with the combined

problem. Furthermore, as an extension of DWMIL, the same authors [44] proposed an adaptive batch-based dynamic weighted majority (ACDWM). The batch size is adaptively increased until the classifier produces stable predictions. Thwart and schenck proposed a two-stage active learning algorithm [45]. Korycki et al. [46] proposed a two-module uncertainty-based active learning strategy for partially labeled nonstationary and imbalanced data streams. This method is only suitable for binary classification. Korycki and krawczyk [47] proposed a solution for CI among multiple classes and CD in the presence of limited labels for multi-class classification. A comprehensive study of the widely used classification and regression techniques was conducted [48]. It is evaluated how well the ensemble approaches perform [49]. The effectiveness of the various sampling approaches is examined [50]. Machine learning methods were compared in this investigation [51]. Table 1 summarizes all the related works.

Wang et al. [3] analyzed the impact of $p(y)$, $p(y/x)$, and $p(x/y)$ drifts independently on active and passive approaches, which are intended to learn from non-stationary environments. Throughout their study, the degree of imbalance is considered as 1:9, and for most cases, it is static. The authors pointed out that in the existence of OCI-CD, the impact of class imbalance in both static and dynamic forms is more critical than the $p(y/x)$ and $p(x/y)$ drifts. But the authors didn't simulate the environment to derive that conclusion on the combined problem of OCI-CD drift. Furthermore, they stated that when three drifts are considered independently, the $p(y/x)$ drift has a critical impact on learner performance concerning $p(y)$ and $p(x/y)$ drifts, and considered drift detection methods are still much susceptible to different types of drifts. However, these observations are intended for online bagging classifiers (i.e., for both active and passive), and the other state-of-the-art adaptive and non-adaptive learners are not considered.

Therefore, to address the above-mentioned gaps, the impact of the training stream's characteristics, including the degree of imbalance, length at the time t , drift types (CI, CD, and OCI-CD), and the state of imbalance (static and dynamic) on state-of-the-art adaptive and non-adaptive learners used for minority class prediction, is explored.

5 Background

This section presents the adaptive and non-adaptive online learning algorithms and evaluation measures used in this study.

5.1 Online learners

The online learning algorithms update the model or learn a new model even when a single sample is available to learn.

(a) Naive bayes [52]:

This algorithm uses the baye's independent assumption on the likelihood conditional probabilities. Initially, the online version predicts the class of each evolving sample with the highest probability among given N classes. This

new sample is used for training to update the probabilities of the existing model. Hence, the model learns incrementally for each evolving sample. For adaptive NB, the required probabilities are calculated only on the window of current training at time t .

$$p(y/x) = \frac{p(x/y)p(y)}{p(x)} \quad (1)$$

(b) Perceptron [52]:

A Perceptron can be learned either online or in batch mode. In an online mode, each sample (input) of size m having the bias b is fed to the neural network which gets initialized with a random weight vector (W). Next, at the level of each neuron, the summation $\sum(WX+B)$ is computed and the output y is predicted by applying activation function f . Each evolving sample undergoes predefined number of epochs until ($Y-O$) becomes minimum by the updation of:

$$W_{new} = W_{old} + \eta(Y - O) x \quad (2)$$

where Y is the target and O is the observed prediction from the model.

(c) KNN [52]:

As it is a lazy learning algorithm, the model is built whenever the test sample invokes it. For adaptive KNN, an initial training set is maintained in a window of constant size. The evolving test samples are predicted against this window of training set by considering the class label of the K of its nearest neighbors. Once after the prediction, the new samples are added to the end of the window. As long as new samples are added to the window, the old knowledge is forgotten. Hence, the window with these data is tending to the current time and can be adaptive to the change. For non-adaptive KNN, the model is built on the entire training set.

(d) VFDT [53]:

A very fast decision tree algorithm, builds a decision tree on evolving data based on hoeffding bounds. The main idea behind this hoeffding bounds is to get some confidence on the data that so far seen. Initially, the root node is fitted over the available data, sufficient statistics are calculated to compute the information gain on each attribute. Let $G(X_a)$ be the attribute a of the highest information gain and $G(X_b)$ be the attribute b of the second highest information gain among the given attributes. If $G(X_a) - G(X_b) > \epsilon$, a split can be carried out on attribute a , and for all branches of the split is replaced with a leaf node which is again initiated with sufficient statistics. Further, newly arriving instances are forwarded to the leaf nodes, where the updation of the model takes place. In this manner, the tree is incrementally updated concerning the newly evolving instances. Here the ϵ is calculated as

$$\epsilon = \sqrt{\frac{R^2 \ln \frac{1}{\delta}}{2n}} \quad (3)$$

where R is the range of a real valued variable r , n is the number of independent observations so far seen, r is the mean of the n independent observations. According to hoeffding bound with $1-\delta$ probability, the true mean of the variable is $r-\epsilon$. For adaptive VFDT the model is built on window of training samples of current time t .

Method	Type of learning		State of imbalance		Drift		Drift Detection	
	Batch	Online	Static	Dynamic	$p(y/x)$	$p(y)$	Active	Passive
[25]	✓		✓		✓		✓	
[26]	✓		✓		✓		✓	
[27]		✓	✓		✓		✓	
[28]	✓	✓		✓	✓	✓	✓	
[29]		✓	✓	✓	✓	✓	✓	
[30]		✓	✓	✓	✓	✓	✓	
[31]	✓		✓		✓		✓	✓
[32]		✓	✓		✓			✓
[33]	✓		✓		✓			✓
[34]	✓		✓	✓	✓	✓		✓
[35]	✓		✓		✓		✓	
[36]	✓		✓		✓		✓	
[37]	✓		✓		✓			✓
[38]		✓	✓	✓		✓		✓
[39]		✓	✓		✓			✓
[40]		✓	✓		✓			✓
[41]	✓	✓	✓		✓		✓	
[42]		✓	✓	✓		✓		✓
[43]	✓		✓		✓			✓
[44]	✓		✓		✓			✓
[45]	✓			✓	✓			✓
[46]		✓	✓		✓			✓
[47]		✓	✓	✓	✓	✓		✓

Table 1: Summary of all the related works.

(e) LASVM active learning [54]:

To accommodate the learning from the incrementally arriving data the standard SVM tools such as LIBSVM [55] are prone to re-training [56], where the QP is repeatedly solved from scratch which is computationally expensive when the data becomes large.

To overcome this problem, a large support vector machine (LASVM) is proposed. This is a kernel-based online active learning algorithm, that *Process* the new sample by trying to add it to the existing support vectors set S and some blatant non-support vectors are removed from S by *Reprocess*. Usually, the active learning methods are incremental and with respect to SVM, the next learning task proceeds from the current boundary. Thus, for each new sample, at the end of *Process* (and *reprocess*) a new SVM boundary is learned.

The QP problem of LASVM simply extends the optimization procedure of sequential minimum optimization (SMO) [57] and computes the gradient from the previous α 's and S . Hence, learning becomes faster. Further, the boundary always models the current data being learned. With the notion of adaptability to the change in data and due to its applicability to imbalanced standalone datasets [58] though, there are much incremental learning approaches [55] for SVM online learning, here LASVM is considered for this study.

As per our knowledge, it is the first sort of study that aims to analyze the behavior of LASVM on $p(y)$, $p(y/x)$, OCI-CD drifts.

5.2 Performance measures

Since the main focus is on the minority class prediction, evaluation prequential [2] of minority class *Recall* is used as the evaluation measure to anticipate the performance. According to [3, 27], *Recall* is used as a performance evaluator. Eventually, the performance of the online learners is depicted with incremental learning curves with the number of instances versus minority class *Recall*.

Evaluation prequential:

It is an interleaved test-and-train procedure to evaluate the data stream by testing each evolving sample on the learned model, after that using it for training.

Classification Recall:

The classification *Recall* of each class usually measured as the

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Here TP referred as number of positive class samples predicted as positives and FN referred as number of positive class samples predicted as negatives. Though the size of the stream is usually assumed to be infinity, for the sake of comparison different sizes are assumed. In this work, the prequential results are reported for 50 fixed chunk sizes. Hence, the total time steps covered for each chunk size is (size of the stream/50).

6 Study design

This section depicts the data generation procedure and the experimental setup used in this study.

6.1 Data sets

Table 2 depicts the synthetic stream generation procedure from two generating functions such as CIRCLE and LINE [59]. From each of these stream generators, two states of imbalance STATIC (i.e., static degree of imbalance) and DYNAMIC (i.e., $p(y)$ the prior probabilities of the classes change dynamically i.e., dynamic degree of class imbalance) are generated as shown in table 3. For each of these states, streams with varying degrees of imbalance such as [1:9, 2:8, 3:7, 4:6, and 5:5] are generated. For each of these imbalanced streams, drifts with three different speeds such as NO (i.e., no drift and the streams are stationary), Gradual (the drift starts at the middle of the stream and it takes few time steps to undergo a complete change in the underlined concept) and Abrupt (the drifts start from the middle of the stream and it takes only one time step to undergo a complete change in the underlined concept) are generated.

To generate Gradual and Abrupt drifts, the speed of the drift is varied from 1 to (chunk size * 10). Each synthetic data set contains a single drift and this drift is simulated with either of these three different severities such as NO (i.e., severity=0%), LOW (i.e., severity≈16%), and HIGH (i.e., severity≈66%) [59] (as shown in table 4). Here, the severity refers to the percentage of change in the underlined concept after the drift. The streams with the specified settings are generated for lengths such as [1K, 50K, 100K, 150K, and 200K]. Here K refer to the size 1000.

Figure 1 depicts the data generation scenario for DYNAMIC imbalance and HIGH Drift for both CIRCLE and LINE generators. In addition to the simulated datasets, a real-world CD dataset KDD CUP 99 [60] is also used in the analysis.

6.2 Experimental setup

Except the LASVM, the online learners of both adaptive and non-adaptive algorithms such as non-adaptive NB, adaptive and non-adaptive KNN, and VFDT, Perceptron is considered from MOA. For non-adaptive KNN the window size is reset for training set sizes. The adaptive NB is implemented in MATLAB. The same window size is considered for both NB and KNN. On the other hand, the LASVM was originally developed to perform SVM active learning on an offline training set incrementally, whereas this has been modified to learn CD streams online. Mainly evaluation prequential of minority class *Recall* is used to demonstrate the performance of the classifiers, and majority class *Recall* is also used where ever necessary. Due to the nature of automatic adjustment of the boundary towards every incoming data both the LASVM and Perceptron are considered adaptive algorithms.

7 Experimental results

This section explores the analysis on the research questions that are already stated. Though we have carried out the proposed study on all aforementioned synthetic streams, the results for small (1K) and large (200K) datasets are presented for simplicity. The Gradual drifts are also used based on the necessity.

RQ1: Does the length of the stream concerning the imbalance ratio at the current time t impact the online learner's performance?

To address this research question, we have considered the case of STATIC Imbalance- NO Drift i.e., the degree of imbalance is static and there is no drift in evolving stream. Here, each stream length [i.e., 1K, 50K, 200K] with varying degrees of imbalances are considered such as [1:9, 2:8, 3:7, 4:6, 5:5]. From Table 5, for all the streams from both generators, on all considered learners, it is identified that as the length of the stream increases the performance also increases until the learning saturates from each evolving stream (Figure 2).

This trend is observed the same for all degrees of imbalance ratios. However, the rate of convergence to maximum recall varies with the degree of imbalance in the stream. From Table 5, for all the streams from both generators, on all considered learners, it is identified that as the degree of imbalance decreases from 1:9 to 5:5, the minority class *Recall* converging rate increases [See horizontally table. 5]. At 1:9 degree of imbalance, the stream is not able to rise from the minimum *Recall* value (i.e., 0), at 5:5 degree of imbalance, the stream performance saturates nearly at maximum *Recall* value (i.e., 100%). This is to be observed same for the streams from both generators, on all classifiers. Although perceptron yielded better performance (i.e., 10 times altogether), its performance is not consistent with the circle stream generator whereas LASVM (7 times altogether) exhibited consistent improvement in its performance with varying degrees of imbalances as well as size.

It is also identified for the balanced streams (i.e., 5:5), the time stamps required to converge to maximum *Recall* also decrease when compared to imbalanced streams. The stream with 5:5 converging to its maximum *Recall* 100%, is below the first 1000-time stamps, whereas the rest of the imbalanced streams converged to their maximum *Recall* after the first 1000 steps. This trend is illustrated with adaptive NB and LASVM (figure 2).

In addition to this, with LASVM, performance improvement is observed compared with the rest of the learners. The [40%, 45%] of the *Recall* yielded by LASVM at high degrees of imbalance [1:9, 2:8] after 1000-time steps, whereas it is observed as zero or nearly zero for the rest of the learners. One of the findings on standalone data sets [61] where the minority class performance is affected by data set size, does not directly implicate imbalanced evolving streams as the length of the stream usually tends to infinity. Here the degree of imbalance only plays a critical role in performance degradation.

Dataset	Imbalance [1:9,2:8,3:7,4:6,5:5]	Speed of the drift	Dataset Sizes	Type of the drift
CIRCLE	STATIC	NO Gradual Abrupt	[1K,50K,100K,150K,200K]	$p(y/x)$
	DYNAMIC	NO Gradual Abrupt	[1K,50K,100K,150K,200K]	$p(y)$, OCI-CD
LINE	STATIC	NO Gradual Abrupt	[1K,50K,100K,150K,200K]	$p(y/x)$
	DYNAMIC	NO Gradual Abrupt	[1K,50K,100K,150K,200K]	$p(y)$, OCI-CD
KDDCUP 99	STATIC [2:8]	-----	500K	$p(y/x)$

Table 2: Synthetic Dataset Description.

State of imbalance	Before	After
STATIC	1:9	1:9
DYNAMIC	1:9	9:1

Table 3: State of imbalance before and after drift for 1:9 case.

Problem	Fixed values	Before- >after drift
CIRCLE	$a=b=0.5$	$r=0.2 \rightarrow 0.2$ (NO DRIFT) $r=0.24 \rightarrow 0.3$ (LOW DRIFT) $r=0.2 \rightarrow 0.5$ (HIGH DRIFT)
LINE	$a1=0.1$	$a0=-0.25 \rightarrow -0.25$ (NO DRIFT) $a0=-0.25 \rightarrow -0.7$ (LOW DRIFT) $a0=-0.1 \rightarrow -0.8$ (HIGH DRIFT)

Table 4: Settings of CD Generators.

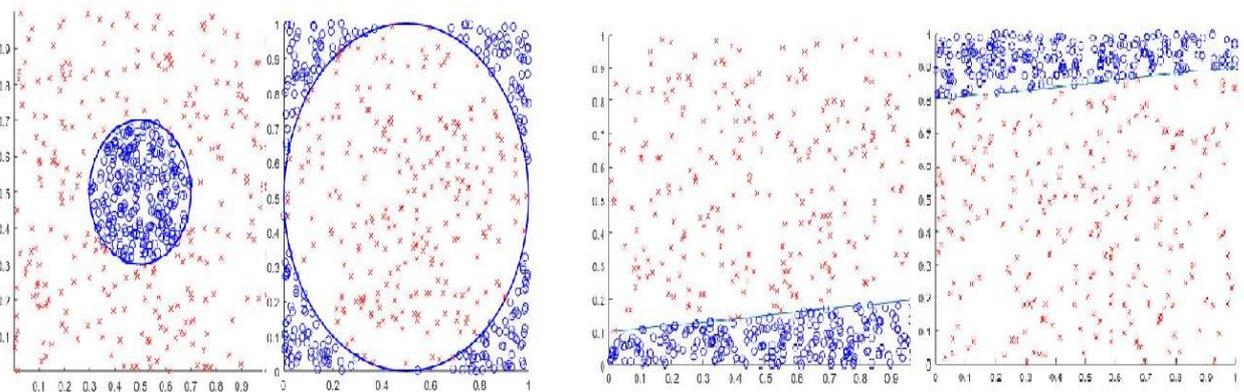


Figure 1: Dynamic imbalance.

RQ2. Is the degree of imbalance or CD whose impact critical in minority class performance degradation?

Here, the main concern is on studying the degree of imbalance and CD on minority class performance. Thus, this question is addressed with STATIC imbalance- HIGH drift case. The degree of imbalance for all streams is considered constant throughout the stream. From figure 3, it is observed that the non-adaptive NB, VFDT, and KNN learners are sensitive to the drift compared to the adaptive versions. At the drift point, the learner’s performance of the minority class started to drop significantly.

This effect is critical for moderate degree of imbalance cases such as [3:7, 4:6, and 5:5] %, due to the low impact of class imbalance on learner’s performance (From RQ1). Besides, for the high degree of imbalance cases such as [1:9, 2:8] %, it is observed that the impact of the degree of imbalance on the learner’s performance is more critical than the impact of the CD. This is due to minimum or zero performance of the learners in terms of minority class *Recall*.

Size	Classifier	Recall (%) for different imbalance ratios				
		1:9	2:8	3:7	4:6	5:5
1K(CIRCLE)	NB (nonadap)	0	19.5	61.66	90.25	98.8
	NB (adap)	5	25	57.66	77.5	89.8
	KNN (nonadap)	6	30	62.66	90.25	99.2
	KNN (adap)	0	11	52	87.25	98.6
	VFDT (nonadap)	0	0	61.66	90.25	98.6
	VFDT (adap)	0	0	67.33	91	91
	Perceptron	0	0	0	25.75	11.6
LASVM	8	14.5	70	91	99.6	
200K(CIRCLE)	NB (nonadap)	0	26.1	63.9	95.4	100
	NB (adap)	0.01	26.8	62.97	94.11	100
	KNN (nonadap)	1.4	24.5	73	98.8	100
	KNN (adap)	2.6	42.5	64.4	95.9	99.8
	VFDT (nonadap)	2.3	0.3	93.3	99.7	99.9
	VFDT (adap)	17.7	85.9	99.8	99.8	99.9
	Perceptron	84.2	92.5	100	100	10.9
LASVM	42.15	45.9	83.95	95.42	100	
1K(LINE)	NB (nonadap)	0	50	81	91.75	98.4
	NB (adap)	12	41.5	70.66	86.75	96.8
	KNN (nonadap)	0	18.5	50.33	87.5	97
	KNN (adap)	0	18.5	50.33	87.5	97
	VFDT (nonadap)	0	1.5	75.66	95.25	97.8
	VFDT (adap)	0	2.5	75.66	95.5	97.2
	Perceptron	0	0	32.33	78	98.4
LASVM	7	20	67	87	98.4	
200K(LINE)	NB (nonadap)	0	57.3	86	96	100
	NB (adap)	0.01	56.35	84.9	94.4	100
	KNN (nonadap)	0.4	17.5	63.7	96.7	100
	KNN (adap)	5.7	49.6	69.8	98	99.7
	VFDT (nonadap)	0.5	1.6	97.2	99.9	99.9
	VFDT (adap)	21.2	20	97.5	95.9	100
	Perceptron	77.1	87.7	100	100	100
LASVM	40.74	43.60	77.81	94.09	100	

Table 5: Minority class prequential *Recall* for CIRCLE and LINE of stream sizes [1K, 200K]. Here nonadap refers to non-adaptive and adap refers to adaptive.

For the adaptive learner such as LASVM, KNN, and NB the effect of the drift seems to be nominal when compared to non-adaptive versions. Though there is an impact of the drift at the beginning of the new concept in terms of a performance drop, the learners are quickly regaining better performances than non-adaptive versions. This trend is observed the same for Gradual cases (figure 4). The adaptability of NB and KNN to the drift is due to the window of the sample's current time t . Whereas LASVM is due to active learning of new samples near to boundary.

RQ3: Is the impact of OCI-CD is more adverse than $p(y)$ and $p(y/x)$ drifts on the learner's performance?

This research question is addressed by considering the cases of DYNAMIC imbalance- NO DRIFT(CI), DYNAMIC imbalance- HIGH DRIFT (OCI-CD). Here each of the stream starts with one of the imbalance degrees (i.e., Minority: Majority) such as [1:9, 2:8, 3:7, 4:6, 5:5] and after the $p(y)$ drift, the degree of imbalance changes to [9:1, 8:2, 7:3, 6:4, 5:5] (i.e., the minority becomes the majority and majority becomes the minority). For simplicity here Class 1 is referred to as minority class and

class 0 is the majority class before the drift. This research question is illustrated with stream size 200K, because the visibility of the drift is vibrant. In case of DYNAMIC imbalance-HIGH Drift both the $p(y)$ and $p(y/x)$ change have tailored to occur at the middle of the stream.

From figure 5, 6, 7, 8 it is noticed that the performance of the two classes hindered by both CI, OCI-CD drifts. However, it is identified that this impact varies from classifier to classifier.

Concerned with Class 1, it is observed that in DYNAMIC imbalance- NO DRIFT case, for the high degree of imbalance cases such as 1:9, 2:8, the performance improvement is observed after the drift point and whereas for moderate degree of imbalance cases such as 3:7 and 4:6, the performance drop down is observed. This trend is similar for both adaptive and non-adaptive cases but more significant in non-adaptive learners (figure 5, 6, 7, 8). On an average the performance of DYNAMIC imbalance - NO DRIFT and DYNAMIC imbalance- HIGH DRIFT are very much similar to each other in adaptive learners.

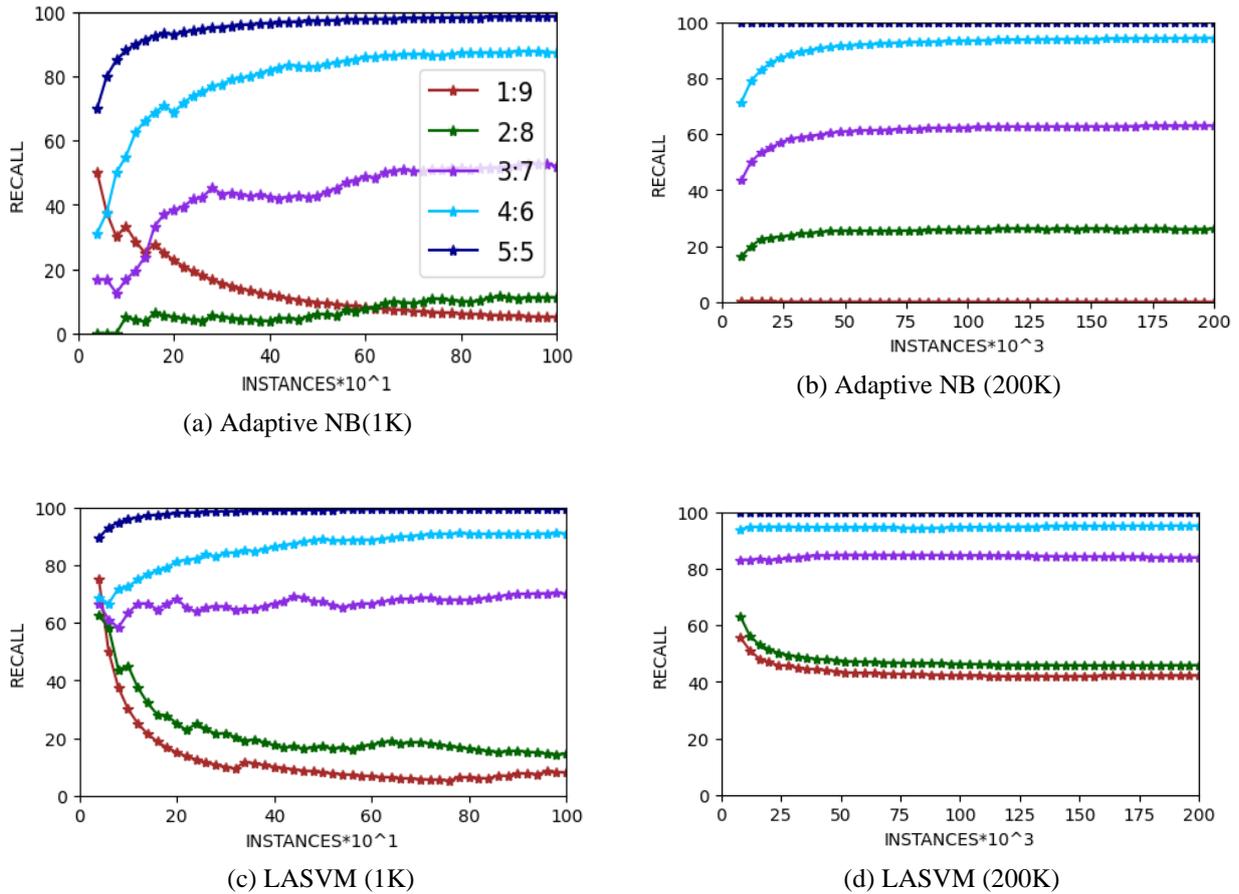


Figure 2: Minority class *Recall* Prequential for the stream sizes [1K, 50K, 200K] on STATIC IMBALANCE-NO DRIFT for CIRCLE DATASET.

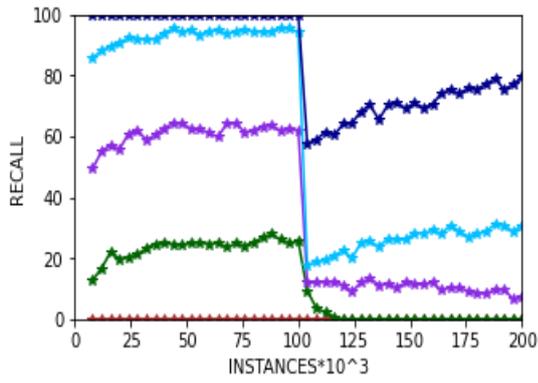
As the both drifts point to the same position it is observed that the impact of $p(y)$ drift is more prominent than $p(y/x)$ drift in OCI-CD (figure 5, 6, 7, 8 of Class 1). Compared to non-adaptive learners, adaptive learners are coping better with CI, OCI-CD drifts.

Concerned with Class 0, with DYNAMIC imbalance- NO DRIFT, after the drift point there is a performance drop down on the streams with most of the learners. But LASVM exhibits a stable performance towards $p(y)$ drift compared to the rest of the learners. However due to either the adaptive nature of the learner or more learning before the drift and scarcity of the concept after, few streams exhibited stable performance. In the case of DYNAMIC imbalance- HIGH drift, besides LASVM rest of the learners have shown performance drop compared to DYNAMIC imbalance-NO drift. In addition to this it is observed that this impact is more on non-adaptive learners when compared with other learners. Based on the learning mechanism the non-adaptive learners are prone to $p(y)$ and OCI-CD drift. Since the perceptron behavior is not consistent in both adaptive and non-adaptive learner’s we cannot derive conclusions (figure 5, 6, 7, 8 of Class 0). Compared to non-adaptive

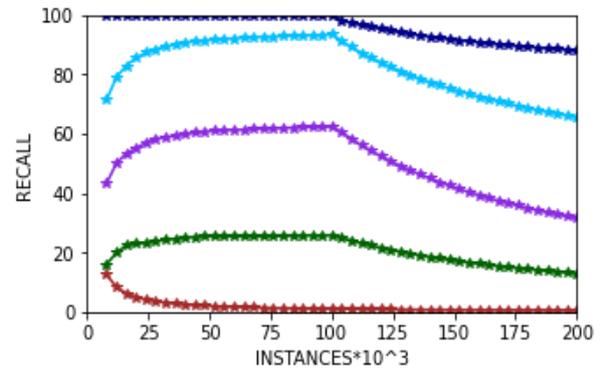
learners, adaptive learners are coping better with CI, OCI-CD drifts.

For non-adaptive NB, KNN, and VFDT, there is a performance drop after the drift. This impact is observed significant for both DYNAMIC Imbalance- NO DRIFT, DYNAMIC imbalance HIGH DRIFT cases. However, the rate of fall of performance at the drift point, the rate of convergence after the drift of DYNAMIC imbalance HIGH DRIFT case only differs with DYNAMIC imbalance- NO DRIFT case. Here the balanced cases 5:5 are even prone to CI, OCI-CD drifts in case of non-adaptive learners. Compared with non -adaptive learners, the class 1 performance of the adaptive learners is better coping with CI, OCI-CD drifts (figure 5, 6, 7 and 8).

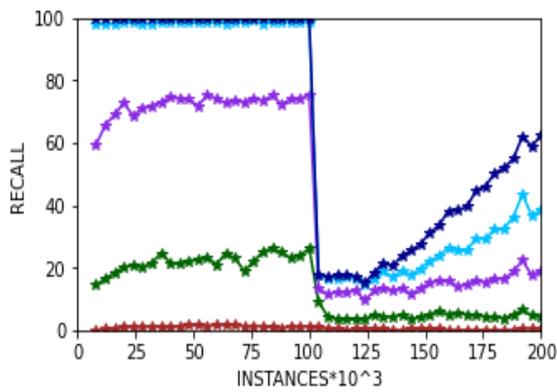
Compared with other adaptive and non-adaptive learners, LASVM exhibits stable performance towards the OCI-CD drift for Class 0. Whereas for class 1, at moderate degree of imbalance cases such as [3:7, 4:6] and at balanced degree of imbalance [5:5] cases are not much sensitive to different types of drifts. However, the high degree of imbalance cases such as [1:9, 2:8] are only prone to CI drift. The same scenario is observed with Gradual drifts on LASVM.



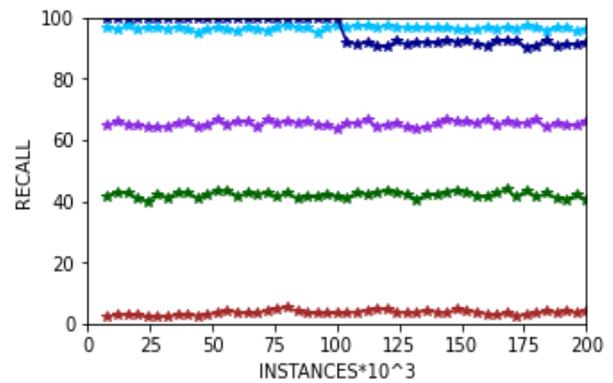
(a) non-adaptive NB (200K)



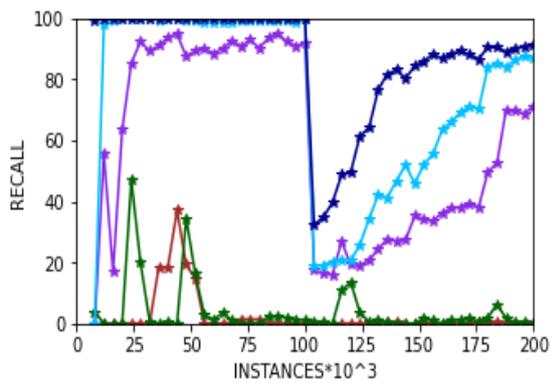
(b) Adaptive NB (200K)



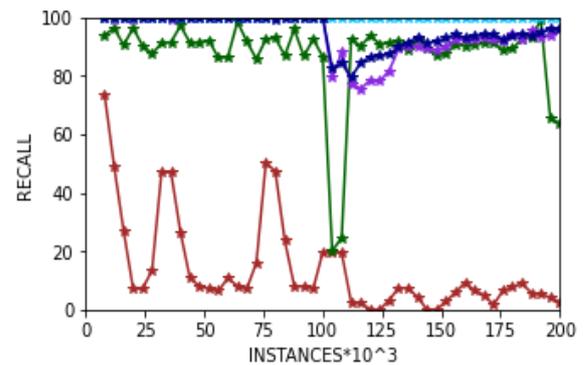
(c) non-adaptive KNN (200K)



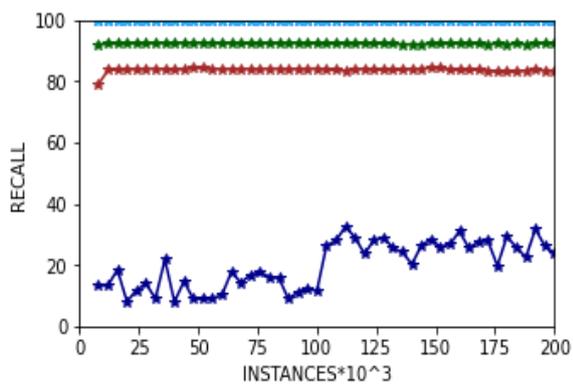
(d) adaptive KNN (200K)



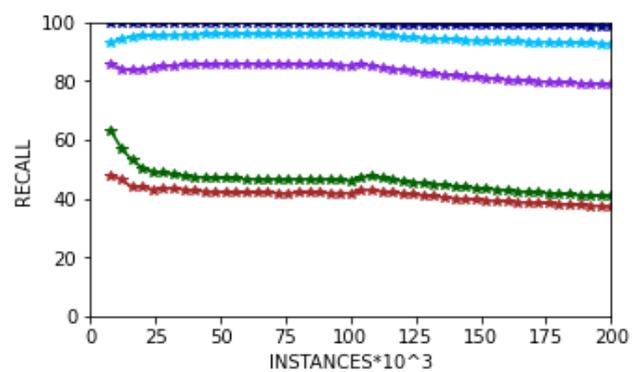
(e) non-adaptive VFDT (200K)



(f) adaptive VFDT (200K)



(g) PERCEPTRON (200K)



(h) LASVM (200K)

Figure 3: Minority class *Recall* Prequential on STATIC IMBALANCE- HIGH DRIFT (Abrupt) for CIRCLE dataset.

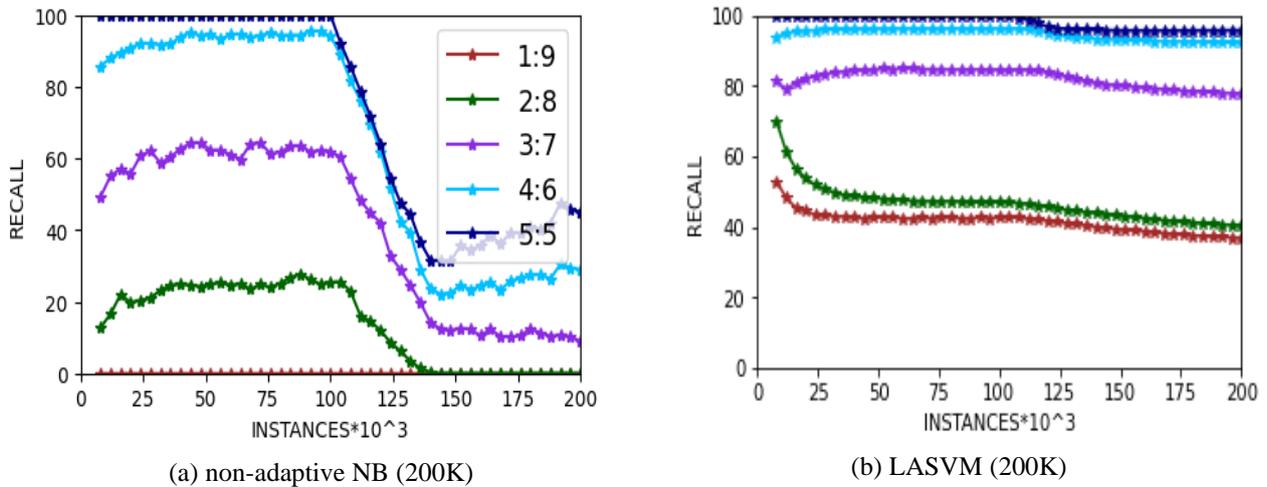


Figure 4: Minority class *Recall* Prequential on STATIC IMBALANCE- HIGH DRIFT (Gradual) for CIRCLE dataset

RQ4. To what extent does online SVM cope with OCI-CD, $p(y)$, and $p(y/x)$ drifts compared to other online learners?

From RQ1, the LASVM has consistently exhibited better performance than the other adaptive and non-adaptive learners, in coping with the degree of imbalance. As the degree of imbalance decreases [1:9, 2:8, 3:7, 4:6, 5:5], the performance increases. Though the size is increasing from 1K to 200K, it is converging earlier to the maximum performance (*Recall*) (table 5 and figure 2).

From RQ2, unlike other adaptive and non-adaptive algorithms, it is not much sensitive to $p(y/x)$ CD. In this case, both LASVM and adaptive KNN consistently yielded better performance (figure 3).

From RQ3, the LASVM is sensitive to $p(y)$ drift at the high degree of imbalance cases on class 1 such as [1:9, 2:8] only. For LASVM, the performance yield looks the same on both DYNAMIC imbalance- NO DRIFT and DYNAMIC imbalance- HIGH DRIFT cases.

Hence the observations regarding online SVM are mimicking the conclusions of [62, 53] which are investigated in static training set settings. Further, apart from the high degree of imbalance cases the performance of online SVM active learning is not that sensitive to $p(y)$, $p(y/x)$, OCI-CD drifts. Compared with other adaptive and non-adaptive learners, adaptive KNN is performing equally in coping with considered three drifts.

8 Analysis on real world data sets

The conclusions are further validated on real world drift dataset such as KDDCUP'99 of 10% [60] which is having a constant degree of imbalance 20:80. Corresponding data characteristics are depicted in table 2. The dataset is well discriminative in nature (figure 9 (b)) and only prone to $p(y/x)$ drift. Here the last five fisher discriminate components are considered to preserve the non-linearity of the data [63], based on the assumption that the non-linear concepts are not much separable. Consequently, the impact of static imbalance (STATIC) with $p(y/x)$ drift in

the context of adaptive and non-adaptive learners is only studied. This scenario is illustrated with 1K-500K. For better readability, the classifiers from MOA (non-adaptive NB, VFDT, adaptive KNN, Perceptron) are considered for comparisons with LASVM.

From figure 9(a), (b) as the length of the stream increases from 0 to 500K the minority class *Recall* increases from 0% to 100% nearly for all classifiers. Here LASVM converges to the maximum *Recall* earlier than other learners. It is identified that NB is more susceptible to drift (figure 9(b) at 350K) compared to other online learners. On the other hand, in the case of $p(y/x)$ drift LASVM is consistent and has exhibited stable performance compared to other adaptive learners KNN, Perceptron, and non-adaptive learners like VFDT and NB. For the smaller stream lengths i.e., 1K (figure 9(a)), experiments were repeated three times and its average evaluation prequential *Recall* is reported.

Hence, from the real-world datasets, it is identified that the adaptive learners better cope with $p(y/x)$ drift compared with non-adaptive learners. In addition to this, it is prominently identified that LASVM is not much sensitive to static class imbalance as well as to $p(y/x)$ drift, which mimics the findings on synthetic streams. Due to the well separability of the data, the impact of class imbalance on other learners is also minimal.

9 Discussion

In this section, observations are discussed on analyzing the impact of the parameters such as static degree of imbalance, stream length, drifts (i.e., $p(y)$, $p(y/x)$, OCI-CD) on adaptive, non-adaptive learners over unbalanced evolving streams. The main observations based on considered parameters are:

- **Degree of imbalance:** In an evolving stream, as the degree of imbalance increases the performance of the minority class decreases. This is observed true for the cases when the degree of imbalance is either static or dynamic.

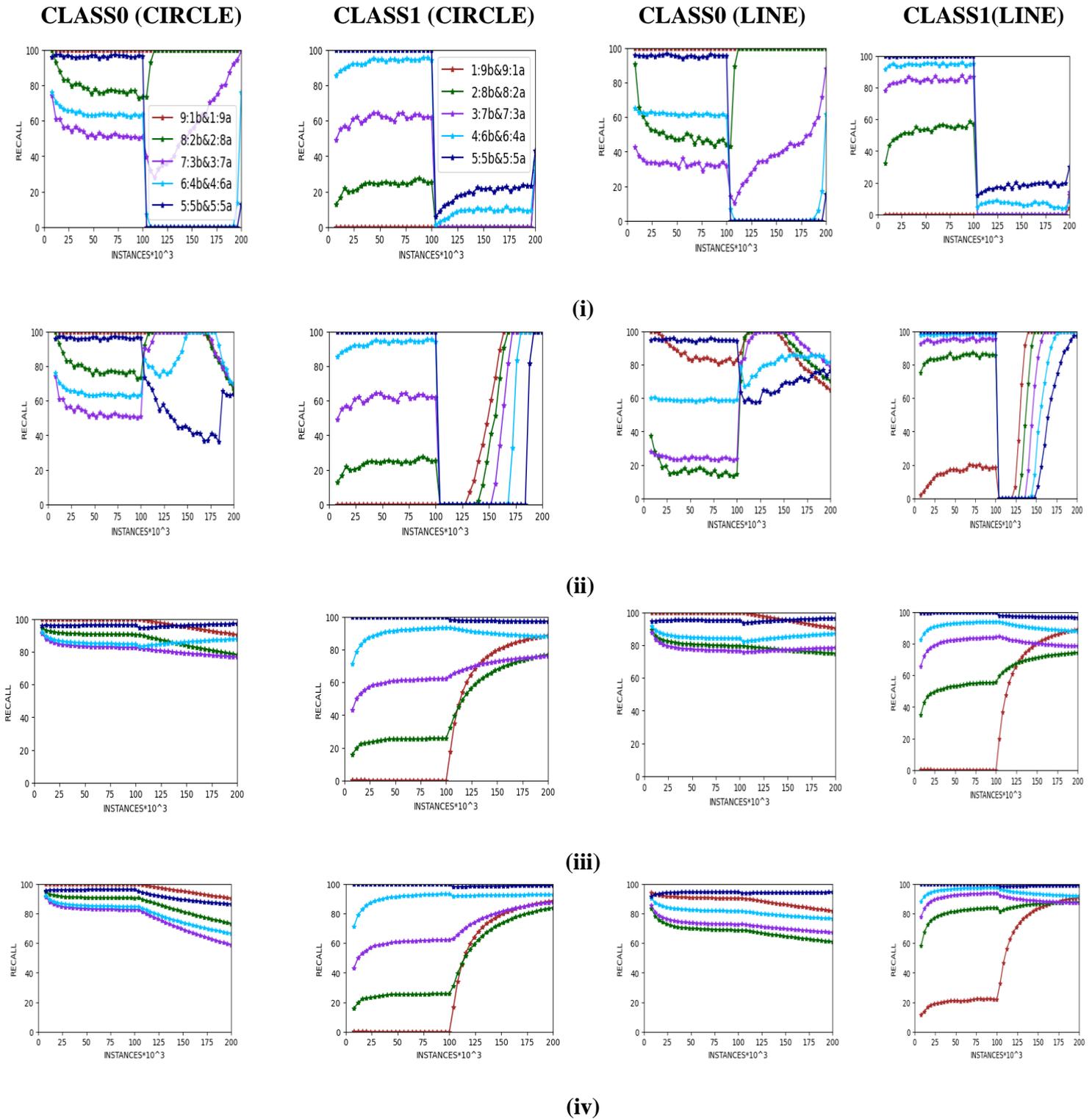


Figure. 5: CLASS 0 and CLASS 1 Recall prequential of DYNAMIC IMBALANCE on NB Learner. Here, b refers to before, a refers to after the drift (Abrupt). (i): non-adaptive NB with no $p(y/x)$ drift. (ii): non-adaptive NB with high $p(y/x)$ drift. (iii): adaptive NB with no $p(y/x)$ drift. (iv): adaptive NB with high $p(y/x)$ drift.

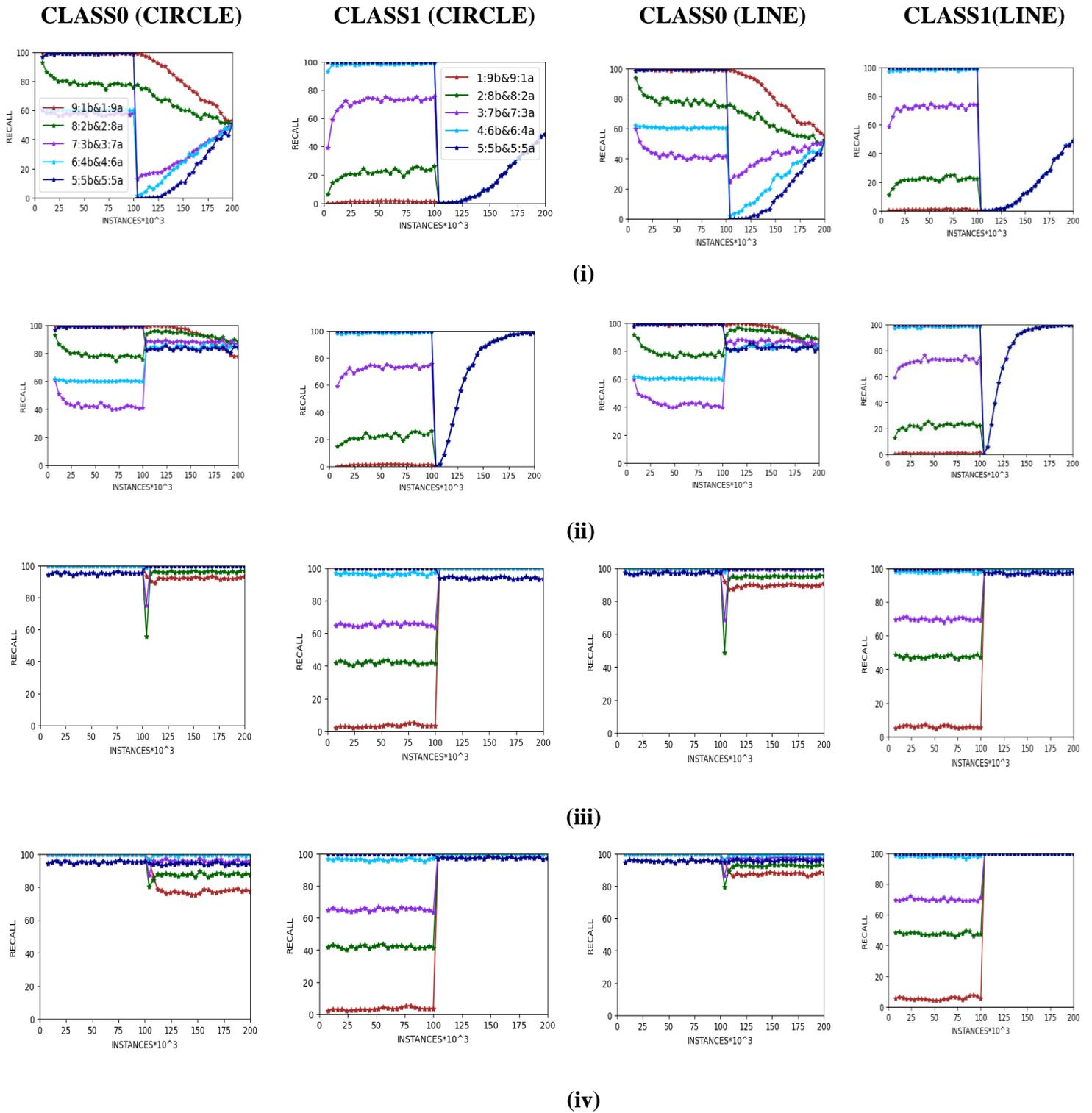


Figure 6: CLASS 0 and CLASS 1 Recall prequential of DYNAMIC IMBALANCE on KNN Learner. Here, b refers to before, a refers to after the drift (Abrupt). (i): non-Adaptive KNN with no $p(y/x)$ drift. (ii): non-adaptive KNN with high $p(y/x)$ drift. (iii): adaptive KNN with no $p(y/x)$ drift. (iv): adaptive KNN with high $p(y/x)$ drift.

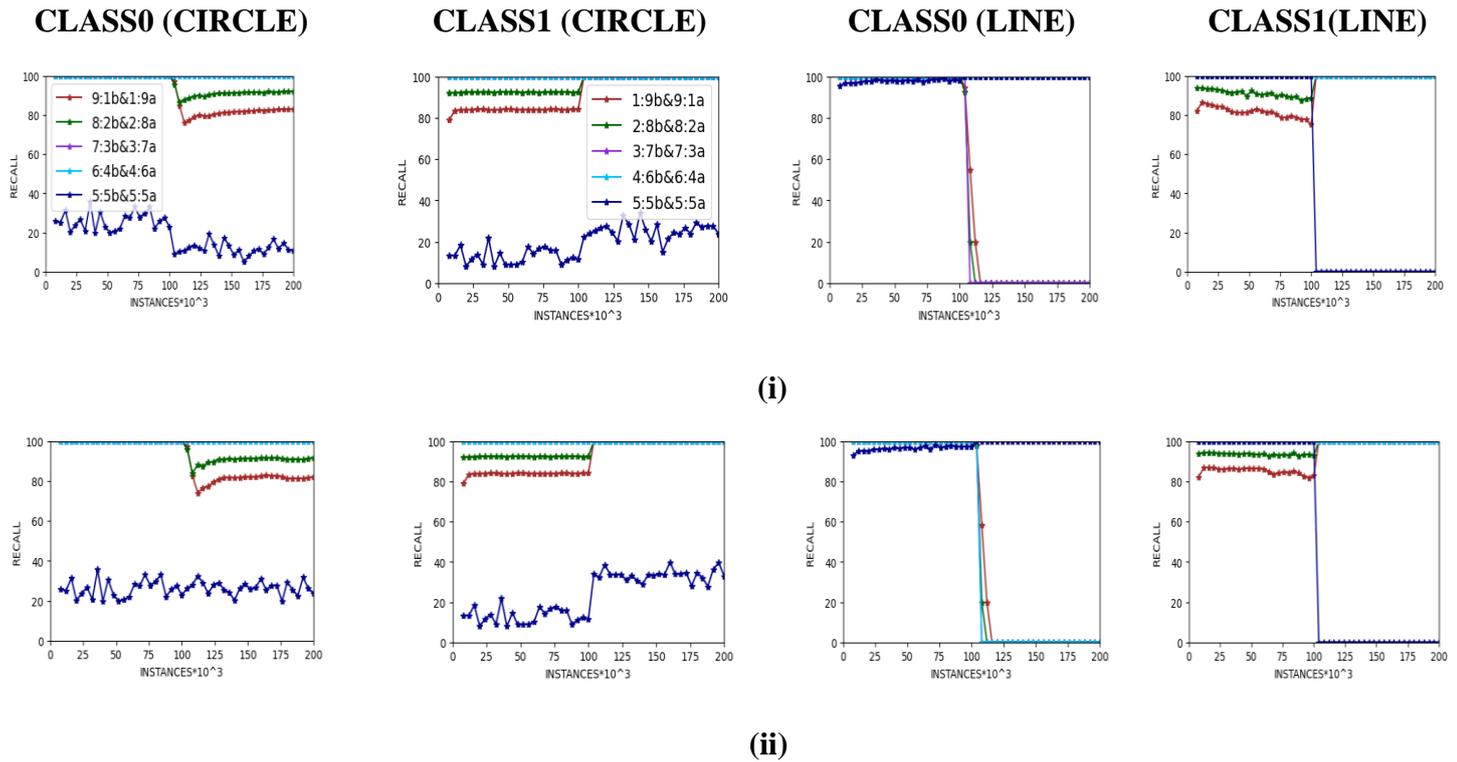


Figure 7: CLASS 0 and CLASS 1 Recall prequential of DYNAMIC IMBALANCE on PERCEPTRON Learner. Here, *b* refers to before, *a* refers to after the drift (Abrupt). (i): Non-Adaptive PERCEPTRON with no $p(y/x)$ drift. (ii): Non-Adaptive PERCEPTRON with high $p(y/x)$ drift.

In the dynamic case, there is a performance drop at the $p(y)$ drift position. This impact is severe for moderately imbalanced streams when compared to the highly. For high imbalance degrees, in a static imbalance state, constant performance degradation is observed. Whereas for dynamic imbalance state, performance improvement is observed after $p(y)$ drift.

- **Length of the Stream:** As the size of the stream increases, the stream with more balanced classes converges earlier than the unbalanced classes. Unlike, on standalone training sets, here the length of the stream with respect to imbalance does not have much impact on minority class performance. However, as the length of the stream increases, the performance increases till the learning from the stream saturates.
- **Real Drift (i.e., $p(y/x)$):** As the length of the stream increases the non-adaptive classifiers are much more prone to $p(y/x)$ drift compared to the adaptive classifiers. This impact is critical for moderate degree of imbalance cases where the imbalance has less impact. For adaptive learners, the impact of the class imbalance on minority class performance is more critical than real drift.
- **Virtual and Real Drift (i.e., $p(y)$ and OCI-CD):** The non-adaptive classifiers are much more prone to both $p(y)$ and OCI-CD drifts compared to adaptive learners. Whereas, the adaptive learners are much more prone to $p(y)$ drifts at the high degree of

imbalance cases. In addition to this, the impact of $p(y)$ and OCI-CD drifts varies from learner to learner and concept to concept. Learners such as NB, KNN, and VFDT are prone to both $p(y)$ and OCI-CD drifts. Further, the impact of dynamic imbalance is more severe than real drift due to the sudden drop down in performances.

- **Adaptive and Non-Adaptive learners:** The non-adaptive learners are much susceptible to drifts (i.e., $p(y)$, $p(y/x)$) and OCI-CD) compared with adaptive learners. The degree of imbalance in two of the forms, static and dynamic is the common factor that both types of learners tend to be prone to. From all considered classifiers apart from the high degree of imbalance cases such as [1:9, 2:8] in both static and dynamic cases, the adaptive large scale SVM active learning is not much sensitive to OCI-CD, $p(y)$ and $p(y/x)$ drift compared to other online learning methods. Further, it is not sensitive to both states of imbalance (static and dynamic). Consistent performances are observed with respect to all considered parameters, though it is not designed to handle CD and $p(y)$ changes in evolving streams.

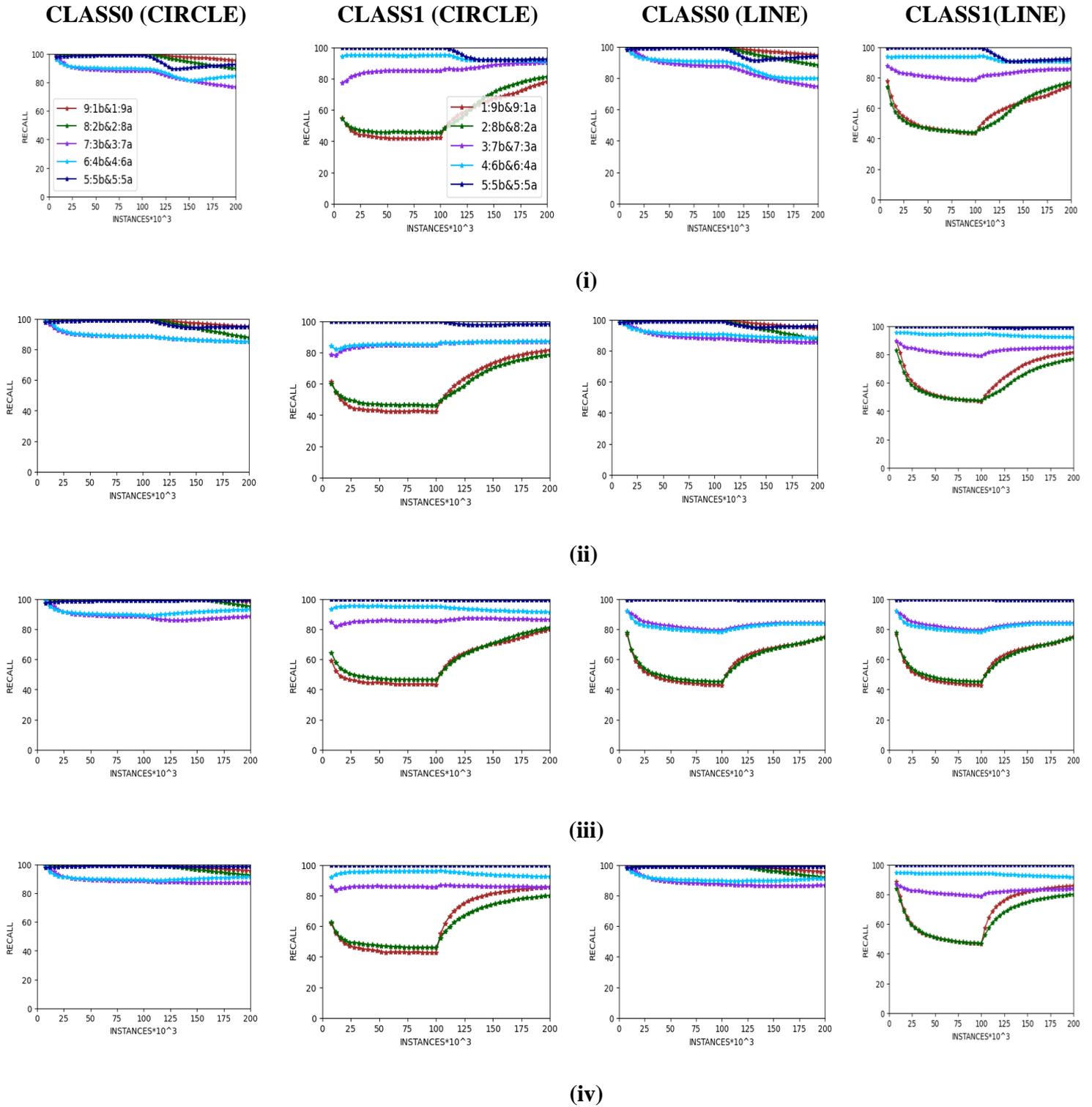


Figure 8: CLASS 0 and CLASS 1 Recall prequential of DYNAMIC IMBALANCE on LASVM Learner. Here, b refers to before, a refers to after the drift. (i): LASVM with no $p(y/x)$ drift (Gradual). (ii): LASVM with high $p(y/x)$ drift (Gradual). (iii): LASVM with no $p(y/x)$ drift (Abrupt). (iv): LASVM with high $p(y/x)$ drift (Abrupt).

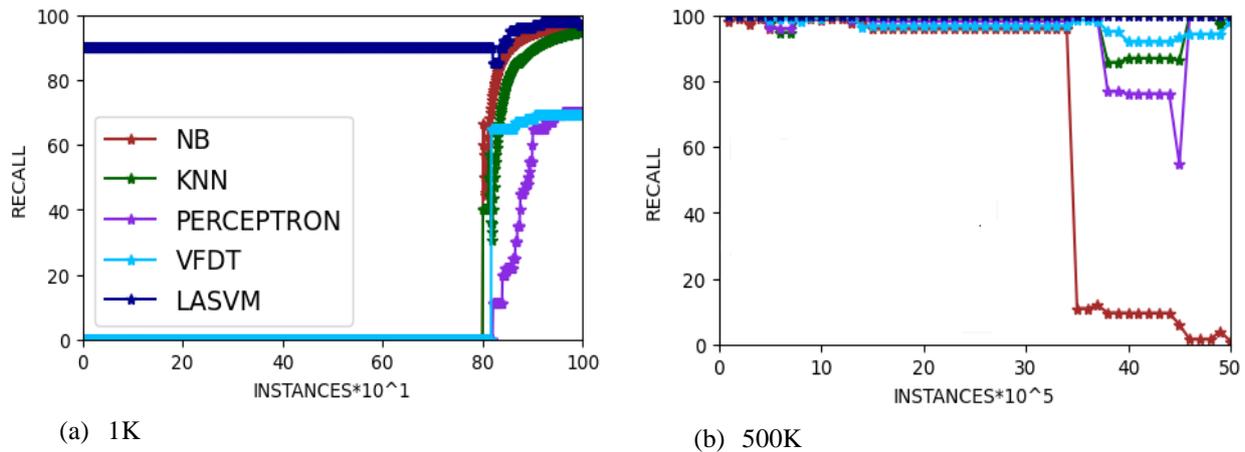


Figure 9 Minority class *Recall* Prequential for KDD CUP from Data Set Size [1K- 500K].

In terms of overall performance, adaptive algorithms outperformed non-adaptive algorithms due to the maintenance of a window throughout the learning process in which older samples are ignored and only representative samples are retained. LASVM outperformed other algorithms in adaptive learners due to its adaptability to changing data and applicability to imbalanced data sets [58]. Furthermore, LASVM employs active learning of new samples near the boundary.

From the above discussions the following recommendations are made to cope with the combined problem of OCI-CD.

- It is visualized that with adaptive methods such as LASVM, which are less sensitive to $p(y/x)$ drifts, the $p(y)$ drift can be handled dynamically by adapting the methods that address the class imbalance problem [4]. This sort of solution is viable for the high degree of imbalanced cases (both static and dynamic) where the impact of $p(y/x)$ drift is nullified by the underperformance of the learner. However, for the environment where tracking of change detection is mandatory, there drift detection methods can be implemented. Whereas the dynamic degree of imbalance is handled with an indicator function [38].
- For non-adaptive learners, the combined OCI-CD problem is approached by employing both drift detection and methods that address the class imbalance problem simultaneously. At first, based on the current degree of imbalance at time t the class imbalance methods are employed. Then the drift can be detected using drift detection methods [1, 2]. If the drift is not detected the current model is updated else a new model is learned with the new sample. However, in case of the dynamic degree of imbalance, the $p(y)$ change could be captured dynamically by an indicator function, and then it can be adaptively countered by the methods that address the class imbalance problem [3]. This sort of solution is viable

for moderate degree of imbalance cases (both static and dynamic) where the tracking of $p(y/x)$ drift is possible and can be addressed.

- New drift detection methods are required to develop for identifying the CI, CD and OCI-CD drifts at the high degree of imbalance cases, or the existing drift detection methods are needed to be fine-tuned for adaptability. However, the change detection methods based on classification error or performance are prone to a state of imbalance in both states of static and dynamic.

10 Conclusion

This work presents an explorative study to analyze the impact of the combined problem of CI (both static and dynamic) and CD (i.e., $p(y/x)$). Initially, this study aims at exploring the impact of the degree of imbalance on online learner's performance. Here it is identified that as the degree of imbalance increases, the performance converged rate of the stream decreases. Further, the balanced streams converging earlier to their maximum performance compared to unbalanced streams. Later, the impact of the CD analyzed over adaptive and non-adaptive learners. It is noticed that the impact of the real CD is more on non-adaptive learners compared with adaptive learners. This effect is critical for an evolving stream with moderate degree of imbalances. For the high degree of imbalance streams, the degree of imbalance is more critical than the CD.

In addition to the above findings, the effects of virtual drift (i.e., $p(y)$) and combined drift (i.e., OCI-CD) are analyzed. It is noticed that the non-adaptive learners such as NB and VFDT are much more prone to both $p(y)$ and OCI-CD drifts. Whereas the adaptive classifiers such as NB, KNN, and VFDT are much more prone to the virtual kind of $p(y)$ drift. Further, to these findings, it is reported that the large-scale active learning SVM, (LASVM) is not

much sensitive to the degree of imbalance as well as different types of drifts though it was not designed for countering the combined problem of virtual and real drift. This study also presents a few guidelines for designing online learning algorithms to address the combined problem of imbalanced evolving streams with CD.

Though the LASVM has better coped with a combined problem compared with other learners, still it is prone to $p(y)$ drift at the high degree of imbalances, therefore an enhanced LASVM for better prediction performance is under study and a drift detector that is able to identify CD in the presence of class imbalance is also under progress.

Acknowledgement

This study was funded by India's defense research and development organization (DRDO) under the sanction code ERIPR/GIA/17-18/038. The work was reviewed by the center for artificial intelligence and robotics (CAIR). We would like to thank the late Dr. T. Maruthi Padmaja for her assistance and support in this work, and she is the grant recipient.

References

- [1] G. Ditzler, M. Roveri, C. Alippi and R. Polikar. Learning in nonstationary environments: A survey. *IEEE Computational intelligence magazine*, 10(4): 12-25, 2015. <https://doi.org/10.1109/mci.2015.2471196>
- [2] J. Gama, I. Zliobaite, M. Pechenizkiy and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4): 1-37, 2014. <https://doi.org/10.1145/2523813>
- [3] S. Wang, LL. Minku and X. Yao. A systematic study of online class imbalance learning with concept drift. *IEEE transactions on neural network learning system*, 29(10): 4802-4821, 2018. <https://doi.org/10.1109/tnnls.2017.2771290>
- [4] H. Haibo, EA. Garcia E. A learning from imbalanced data, *IEEE transactions on knowledge and data engineering*, 21(9): 1263-1284, 2009. <https://doi.org/10.1109/tkde.2008.239>
- [5] Y. Sun, A. Wong and MS. Kamel M S. Classification of imbalanced data: a review. *International journal of pattern recognition and artificial intelligence*, 23(4): 687-719, 2009. <https://doi.org/10.1142/s0218001409007326>.
- [6] K. Morik, P. Brockhausen and T. Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *proceedings of the 16th international conference on machine learning ICML: 268-277*, 1999.
- [7] C. Elkan. The foundations of cost sensitive learning, In *proceedings of intelligence joint conference on artificial intelligence (IJCAI'01): 973-978*, 2001.
- [8] X. Liu and Z. Zhou. The influence of class imbalance on cost-sensitive learning: An empirical study. In *sixth international conference on data mining (ICDM'06): 970- 974*, 2006. <https://doi.org/10.1109/icdm.2006.158>
- [9] HJ. Lee and S. Cho. The novelty detection approach for difference degrees of class imbalance. In I. King, J. Wang, LW. Chan, D. Wang, ed, *neural information processing, ICONIP 2006, 4233*, lecture notes in computer science. springer, berlin, Heidelberg: 21-30, 2006. https://doi.org/10.1007/11893257_3
- [10] W. Fan, SJ. Stolfo, J. Zhang and PK. Chan. Adacost: misclassification cost sensitive boosting. In *proceedings of 16th international conference on machine learning, morgan Kaufmann: 97–105*, 1999.
- [11] Y. Sun, MS. Kamel, AK. Wong and Y. Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern recognition*, 40(12): 3358-3378, 2007. <https://doi.org/10.1016/j.patcog.2007.04.009>
- [12] MV. Joshi, V. Kumar and RC. Agarwal. Evaluating boosting algorithms to classify rare classes: comparison and improvements. In *proceedings 2001 IEEE international conference on data mining*, 257-264. <https://doi.org/10.1109/icdm.2001.989527>
- [13] S. Wang and X. Yao. Diversity analysis on imbalanced data sets by using ensemble models. In *IEEE symposium on computational intelligence and data mining CIDM '09*, 324-331, 2009. <https://doi.org/10.1109/cidm.2009.4938667>
- [14] X. Liu, J. Wu J and Z. Zhou Z. Exploratory under sampling for class imbalance learning. *IEEE transactions on systems, man and cybernetics, part B(cybernetics)*, 39(2): 539-550, 2009. <https://doi.org/10.1109/tsmcb.2008.2007853>

- [15] C. Seiffert, TM. Khoshgoftaar, JV. Hulse and A. Napolitano. RUSBoost: A hybrid approach to alleviating class imbalance, *IEEE transactions on systems, man, and cybernetics - part A: systems and humans*, 40(1): 185-197, 2010.
<https://doi.org/10.1109/tsmca.2009.2029559>
- [16] NV. Chawla, A. Lazarevic, LO. Hall and KW. Bowyer. SMOTEBoost: Improving prediction of the minority class in boosting. In N. Lavrac, D. Gamberger, L. Todorovski and H. Blockeel ed, *Knowledge discovery in databases: PKDD 2003*, 2838, lecture notes in computer science, springer, berlin, heidelberg, 107-119, 2003.
https://doi.org/10.1007/978-3-540-39804-2_12
- [17] H. Guo and HL. Viktor H L. Learning from imbalanced data sets with boosting and data generation: the Data Boost-IM approach. *ACM SIGKDD explorations newsletter - special issue on learning from imbalanced datasets homepage archive*, 6(1): 30-39, 2004.
<https://doi.org/10.1145/1007730.1007736>
- [18] S. Hido, H. Kashima and Y. Takahashi. Roughly balanced bagging for imbalanced data, 2: 412-426, 2009.
<https://doi.org/10.1002/sam.10061>
- [19] H. Blaszczynski and J. Stefanowski. Neighborhood sampling in bagging for imbalanced data. *Neurocomputing*, 184-203, 2015.
- [20] I. Khamassi, MS. Mouchaweh, M. Hammami and K. Ghedira. Discussion and review on evolving data streams and concept drift adapting. *Evolving Systems*, 9: 1-23, 2018.
<https://doi.org/10.1007/s12530-016-9168-2>
- [21] JP. Patist. Optimal window change detection. In seventh IEEE international conference on data mining workshops (ICDMW 2007), 557-562, 2007.
<https://doi.org/10.1109/icdmw.2007.9>
- [22] K. Nishida and K. Yamauchi. Detecting concept drift using statistical testing. In V. Corruble, M. Takeda and E. Suzuki E ed, *Discovery Science*, 4755, lecture notes in computer science, Springer, Berlin, Heidelberg, 264-269, 2007.
https://doi.org/10.1007/978-3-540-75488-6_27
- [23] DM. Hawkins, Q. Peihua and WK. Change. The change point model for statistical process control. *Journal of quality technology*, 35(4): 355-366, 2003.
- [24] A. Wald. Sequential tests of statistical hypotheses. In S. Kotz and NL. Johnson ed. *Breakthroughs in statistics*, springer series in statistics (perspectives in statistics), new york, NY, 1992
https://doi.org/10.1007/978-1-4612-0919-5_18
- [25] S. Micevska, A. Awad and S. Sakr. SDDM: An interpretable statistical concept drift detection method for data streams. *Journal of intelligent information systems*, 56: 459-484, 2021.
<https://doi.org/10.1007/s10844-020-00634-5>
- [26] P. Li, Wu. Man, He. Junhong and Hu. Xuegang. Recurring drift detection and model selection-based ensemble classification for data streams with unlabeled data. *New generation computing*, 39: 341-376, 2021.
- [27] S. Wang, LL. Minku, D. Ghezzi, D. Caltabiano, P. Tino and X. Yao. Concept drift detection for online class imbalance learning. In international joint conference on neural networks, 1-8, 2013.
<https://doi.org/10.1109/ijcnn.2013.6706768>
- [28] H. Wang, Z. Abraham. Concept drift detection for streaming data. In international joint conference of neural networks, 1-9, 2015.
<https://doi.org/10.1109/ijcnn.2015.7280398>
- [29] D. Brzezinski and J. Stefanowski. Prequential auc for classifier evaluation and drift detection in evolving data streams. *New frontiers in mining complex patterns*, 8983, 87-101, 2015.
https://doi.org/10.1007/978-3-319-17876-9_6
- [30] S. Wang and LL. Minku. AUC estimation and concept drift detection for imbalanced data streams with multiple classes. In 2020 international joint conference on neural networks (IJCNN), 1-8, 2020.
- [31] MM. Idrees, LL. Minku, F. Stahl and A. Badii. A heterogeneous online learning ensemble for non-stationary environments. *Knowledge-based systems*, 188, 104983, 2020.
<https://doi.org/10.1016/j.knosys.2019.104983>
- [32] J. Gao, W. Fan, J. Han and P. Yu, P. A general framework for mining concept drifting data streams with skewed distributions. In proceedings of the 2007 SIAM international conference on data mining (SDM), 3-14, 2007.
<https://doi.org/10.1137/1.9781611972771.1>
- [33] S. Chen and He. Haibo. SERA: Selectively recursive approach towards nonstationary imbalanced stream data mining. In 2009 international joint conference on neural networks, 522-529, 2009.
<https://doi.org/10.1109/ijcnn.2009.5178874>
- [34] R. Lichtenwalter and NV. Chawla. Adaptive methods for classification in arbitrarily imbalanced and drifting data streams. In T. Theeramunkong et al ed. *new frontiers in applied data mining, PAKDD*, 5669, Springer, Berlin, Heidelberg, 53-75, 2009.
https://doi.org/10.1007/978-3-642-14640-4_5

- [35] T. Ryan Hoens and NV. Chawla. Learning in non-stationary environments with class imbalance. In 18th ACM SIGKDD international conference on knowledge discovery and data mining, 168-176, 2012.
<https://doi.org/10.1145/2339530.2339558>
- [36] TR. Hoens, NV. Chawla, R. Polikar. Heuristic updatable weighted random subspaces for non-stationary environments. In 2011 IEEE 11th international conference on data mining, 241-250, 2011.
<https://doi.org/10.1109/icdm.2011.75>
- [37] G. Ditzler and R. Polikar. Incremental learning of concept drift from streaming imbalanced data. IEEE transactions on knowledge and data engineering, 25(10): 2283-2301, 2013.
<https://doi.org/10.1109/tkde.2012.136>
- [38] S. Wang, LL. Minku and X. Yao. Resampling-based ensemble methods for online class imbalance learning. IEEE transactions on knowledge and data engineering, 27(5): 1356-1368, 2015.
<https://doi.org/10.1109/tkde.2014.2345380>
- [39] A. Ghazikhani, R. Monsefi and YH. Sadoghi. Recursive least square perceptron model for non-stationary and imbalanced data stream classification. Evolving Systems, 4: 119–131, 2013.
<https://doi.org/10.1007/s12530-013-9076-7>
- [40] A. Ghazikhani, R. Monsefi and YH. Sadoghi. Online neural network model for nonstationary and imbalanced data stream classification. International journal of machine learning and cybernetics, 5(1): 51-62, 2014.
<https://doi.org/10.1007/s13042-013-0180-6>
- [41] B. Mirza, Z. Lin Z and N. Liu N. Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift. Neurocomputing, 149: 316-329, 2015.
<https://doi.org/10.1016/j.neucom.2014.03.075>
- [42] S. Barua, MM. Islam and K. Murase K. GOSIL: A generalized over-sampling based online imbalanced learning framework. In S. Arik, T. Huang T, W. Lai and Q. Liu ed, Neural information processing, ICONIP, Lecture notes in computer science, springer, cham, 9489, 2015.
https://doi.org/10.1007/978-3-319-26532-2_75
- [43] LU. Yang, Y. Cheung and Y. Tang Y. Dynamic weighted majority for incremental learning of imbalanced data streams with concept drift. In proceedings of the twenty-sixth international joint conference on artificial intelligence, 2393-2399, 2017.
<https://doi.org/10.24963/ijcai.2017/333>
- [44] LU. Yang, Y. Cheung and Y. Tang Y. Adaptive chunk-based dynamic weighted majority for imbalanced data streams with concept drift. IEEE transactions on neural networks and learning systems, 31(8): 2764-2778, 2020
<https://doi.org/10.1109/tnnls.2019.2951814>
- [45] A. Tharwat and W. Schenck W. Balancing exploration and exploitation: a novel active learner for imbalanced data. Knowledge based systems, 210, 2020.
<https://doi.org/10.1016/j.knsys.2020.106500>
- [46] L. korycki, A. Cano and B. krawczyk B. Active learning with abstaining classifiers for imbalanced drifting data streams. In 2019 IEEE international conference on big data (big data), 2334-2343, 2019.
<https://doi.org/10.1109/bigdata47090.2019.9006453>
- [47] L. Korycki and B. Krawczyk. Online oversampling for sparsely labeled imbalanced and non-stationary data streams. In 2020 international joint conference on neural networks (IJCNN), 1-8, 2020.
<https://doi.org/10.1109/ijcnn48605.2020.9207118>
- [48] M. Jena and D. Satchidananda. Decision tree for classification and regression: a state-of-the art review, informatica, 44: 405-420, 2019.
<https://doi.org/10.31449/inf.v44i4.3023>
- [49] R. Saifan, K. Sharif, M. Abu-Ghazaleh and M. Abdel-Majeed. Investigating algorithmic stock market trading using ensemble machine learning methods. Informatica, 44: 311-325, 2020.
<https://doi.org/10.31449/inf.v44i3.2904>
- [50] EB, Francis, Q. Zhen, K. Hughes-Lartey and EA. Kwame. Predicting fraud in mobile money transactions using machine learning: the effects of sampling techniques on the imbalanced dataset. Informatica, 45: 45-46, 2021.
<https://doi.org/10.31449/inf.v45i7.3179>
- [51] AA. Ali and AS. Fakhreldeen. A Comparative analysis of machine learning algorithms to build a predictive model for detecting diabetes complications. Informatica, 45: 117-125, 2021.
<https://doi.org/10.31449/inf.v45i1.3111>
- [52] A. Bifet, G. Holmes, R. Kirkby and B. Pfahringer. MOA: Massive online analysis. Journal of Machine Learning Research, 11: 1601-1604, 2010.
- [53] P. Domingos and G. Hulten. Mining highspeed data streams. In proceedings of 6th ACM SIGKDD international conference on knowledge discovery data mining, Boston, MA USA, 71-80, 2000.
<https://doi.org/10.1145/347090.347107>
- [54] Bordes A, Ertekin S, Weston J, Bottou L (2005) Fast Kernel Classifiers with Online and Active Learning, Journal of Machine Learning Research, 6, pp. 1579-1619.
- [55] C. Chang and CJ. Lin. LIBSVM A library for support vector machines. ACM transactions on intelligent systems and technology, 2(3), 2011.
- [56] L. Pierre-Xavier. Adaptive machine learning algorithms for data streams subject to concept drifts, Ph. D Thesis., University pierre at marie curie, Paris VI, 2017.
- [57] JC. Platt J C. Sequential minimum optimization: A fast algorithm for training support vector machines. A technical report MSR-TR-98-14, 1998.
- [58] S. Ertekin, J. Huang and C. Lee Giles. Active learning class imbalance Problem, In conference on research and development in information retrieval, netherlands, 823- 824, 2007.

<https://doi.org/10.1145/1277741.1277927>

- [59] LL. Minku, AP. White A P and X. Yao X. The impact of diversity on on-line ensemble learning in the presence of concept drift. *IEEE transactions on knowledge and data engineering*, 22(5): 730-742, 2010.
<https://doi.org/10.1109/tkde.2009.156>
- [60] D. Dua and K. Taniskidou. UCI machine learning repository.
<http://kdd.ics.uci.edu/databases/kddcup99.html>
- [61] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5): 429-449, 2002.
<https://doi.org/10.3233/ida-2002-6504>
- [62] TS. Sethi and M. Kantardzic M. On the reliable detection of concept drift from streaming unlabeled data. *Expert systems with applications: an international journal*, 82(c): 77-99, 2017.
<https://doi.org/10.1016/j.eswa.2017.04.008>
- [63] RC. Prati, GEAPA. Batista and MC. Monard. Class imbalances versus class overlapping: An analysis of a learning system behavior. In R. Monroy, G. Arroyo-Figueroa, LE. Sucar and H. Sossa ed, *MICAI: Advances in Artificial Intelligence*, 2972, Lecture notes in computer science, springer, Berlin, Heidelberg, 312-321, 2004.
https://doi.org/10.1007/978-3-540-24694-7_32

