Date of publication DEC-31, 2022, date of current version OCT-27, 2022.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.21.4.2778

An Enhanced Online Boosting Ensemble Classification Technique to Deal with Data Drift

RUCHA C. SAMANT, SUHAS PATIL

Computer Engineering Department, College of Engineering, Bharati Vidyapeeth Deemed to be university, Pune, Maharashtra, India

Corresponding author: Rucha C. Samant (e-mail: ruchasamant25@gmail.com).

ABSTRACT Over the last two decades, big data analytics has become a requirement in the research industry. Stream data mining is essential in many areas because data is generated in the form of streams in a wide variety of online applications. Along with the size and speed of the data stream, concept drift is a difficult issue to handle. This paper proposes an Enhanced Boosting-like Online Learning Ensemble Method based on a heuristic modification to the Boosting-like Online Learning Ensemble (BOLE). This algorithm has been improved by implementing a data instance that retains the previous state policy. During the boosting phase of this modified algorithm, the selection and voting strategy for an instance is advanced. Extensive experimental results on a variety of real-world and synthetic datasets show that the proposed method adequately addresses the drift detection problem. It has outperformed several state-of-the-art boosting-based ensembles dedicated to data stream mining (statistically). The proposed method improved overall accuracy by 1.30 percent to 14.45 percent when compared to other boosting-based ensembles on concept drifted datasets.

KEYWORDS Boosting; concept drift; drift detectors; data stream mining; ensemble classification.

I. INTRODUCTION

Normally the goal of data analysis is to uncover hidden information. In many cases, this knowledge is useful for improving the working model or avoiding hazardous situations. The meaning of data has changed significantly over the last few decades as the sources of data generation have shifted from static to dynamic. The rate of data generation has increased exponentially as a result of the widespread use of online applications and devices, and the nature of data has also changed. Data is now referred to as a data stream, and one of its most significant characteristics is time. As a result, it continues to change over time, becoming increasingly critical in assessing.

Since this data is in the form of streams, it has unique properties that make interpreting and using various machine learning techniques inappropriate. The following are the main data stream processing challenges that need to be resolved, according to investigations [1, 2] on data stream mining literature:

1. The speed with which data streams are created implies that processing the complete data set may be delayed, or may be impossible.
2. Since the size of the stream is so large, it requires more memory than simple applications.
3. Concept drift happens when the underlying data distribution changes in the stream. Because the data is not static, the analysis of each and every example across time may alter. As a result, each slot's prediction changes with time, and the model constructed for such data must be versatile.

Classification is the foundation for analysis in most applications, such as market trends based on customer demands, spam mail rectification, climate change, and share market, to name a few [3]. However, an adaptable algorithm is needed for classifying data streams because of the enormous speed and amount of the data as well as the changing behavior of the data. Traditional single classifiers such as SVM [4] and Decision trees [5] are not sufficient to deal with all of these challenges. Advanced methods, such as adaptive sliding windows, data set sampling algorithms, drift detectors, and adaptive ensembles, have been developed in the literature to face this issue [6–14]. Ensemble classifiers outperformed all of these methods. Similarly, to process huge amount of data, more memory is required. Despite these drawbacks, ensemble has demonstrated its ability to deal with vivid data, has achieved high accuracy, and its simple design methods have drawn the attention of many researchers[1, 2, 15].

Either the short-term or long-term behaviour of a growing data stream may be more relevant in the categorization process and it is sometimes impossible to predict which one is more important in advance [16]. Expert systems must constantly adjust to the new distribution in scenarios where multiple changes of concepts occur. If the time it takes to train the model is longer, the concept may change, and the developed model will become obsolete for new situations. The researchers and academicians made a lot of efforts to solve the problems with data stream classification using ensemble methods based on online and block based, bagging and boosting concepts [17].

Learning ensembles for data streams with idea drifts have been the subject of many studies, primarily to aid in the application of various strategies based on the instructional considerations that are covered in it. To better use ensemble diversity and address idea drifts, several specifications are explored in [18]. These essential considerations make it possible to quantify ensemble diversity and, as a result, produce novel solutions. According to the findings of this study, boosting is a simple but effective method for achieving high accuracy by allowing each base expert to work on weighted data, and the results are produced by a cumulative policy.

In order to categorize data streams with various drift patterns, this paper provides an enhanced boosting ensemble algorithm in conjunction with a drift detector.

Fig. 1 depicts the boosting-based ensemble construction with a standard drift identifier. This research seeks to enhance boosting ensemble classification method for data stream mining. Standard methods like classification accuracy and correctness (kappa) requirements are used to assess the suggested algorithms. An experimental comparison of known and available boosting classification techniques with proposed Enhanced Boosting-like Online Learning Ensemble is conducted in this research.

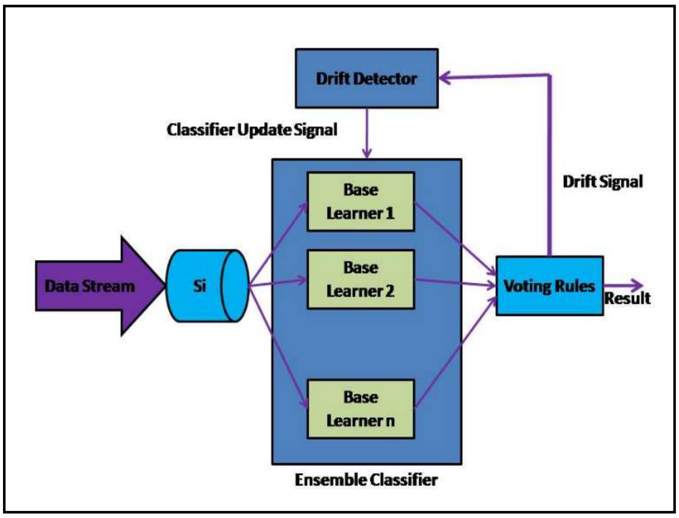

Figure 1. Conventional Ensemble Classifier with Drift Detector

The remainder of the paper is structured as follows. An introduction of data stream mining and associated research in ensemble learning and boosting based classification for data streams are presented in Section 2. A thorough explanation of the proposed Ensemble algorithm's design and guiding principles may be found in Section 3. A complete experimental examination on a sizable number of data streams, including streams with concept drift, is presented in Section 4. Section 5 concludes by summarizing the last thoughts and indicating potential future research directions.

II. LITERATURE REVIEW

Ensemble methods are a powerful technique for improving the precision and robustness of single models: Many classifiers are combined to form an ensemble, and it is expected that the ensemble will perform better at classification than a single method. The conventional techniques of boosting and bagging are employed to increase the accuracy of other algorithms (weak learners). To provide various randomly generated bootstrap samples for training, bagging adopts resampling from the training set with repetitions. In contrast, Boosting modifies the variety provided to train each member based on prior expectations as a way to produce various distributions over the training data [13].

Among the many ensemble methods proposed by researchers, the Online Bagging and Boosting method proposed by Oza and Russell [19] is a watershed moment in the history of data stream classification. In this, input is paired in the online bagging method for training set Z(x, y). The number of iterations, as well as the machine learning method M, is fixed. The machine learning technique used may be a decision tree, SVM, random forest, etc. A bootstrap sample of the same size N is drawn from training set Z for each iteration, and a learning method M is applied to it. Model L is determined by each base classifier (Bc). After repeating the process k times, it will add the sample (xi, yi) to the training set and update all of the base classifiers, returning the ensemble L←(Bc1, Bc2, …..Bcn). The majority vote of all predictors is used to classify the data stream.

The Online Boosting algorithm [19] operates on instances generated using the Poisson distribution. If there is a change in samples, it updates the subsequent base classifier (Bc). The training instances possess weigh ts that are allotted to base experts Bc in the following order: Bc1, Bc2,....Bcn. If the base expert Bc1 misclassifies the training instances, half of the total weight is assigned to the misclassified instances for the next training set, while the remaining half weight is assigned to the correctly classified instances.

Carvalho Santos et al. proposed the Adaptable Diversity-based Online Boosting (ADOB) [20] algorithm, which is a modified version of the online AdaBoosting method that is primarily concerned with the problem of abrupt and frequent concept drift. This method distributes instances more efficiently among experts, allowing it to adapt to concept drifts more quickly. This is accomplished by decreasing the value of poison distribution (λ) when the classification is correct and increasing it when it is incorrect. It has used adaptive window mechanism (ADWIN) to detect drift. The authors contend that this tactic generally tends to increase the ensemble's accuracy following idea drifts, particularly when these drifts are sudden. Last but not the least, ADOB, like the majority of boosting techniques, only permits a classifier (weak hypothesis) to vote if its error is up to 50% and ceases voting when one member violates this requirement. Thus, ADOB technique improves classifier accuracy while reducing execution time and memory usage.

Barros et al. improved ADOB and named it Boosting-like Online Learning Ensemble (BOLE) [21]. All previously designed online boosting methods were based on the same classifier selection criterion, which was that the error should be less than a certain threshold value (below 50 percent). This condition removes moderately behaving classifiers directly. Another disadvantage of the previous system is that it removes incorrectly classified instances, which is an ineffective strategy in an online environment where each instance has an impact on the outcome. In the case of BOLE, there are two different parameters "breakVotes" and "errorBound" that restrict classifiers for voting and avoid negative voting by using the "weightShift" parameter, which has a range of 0.0 to 5.0.

Pelossof et al. created an Online Coordinate Boosting algorithm (OCBoost) that modifies Oza and Russell's algorithm [22] by cancelling weak hypothesis selection criteria. OCBoost uses approximated product term as reference instead of the previous technique of storing margins of all previous instances. Using the standard weight policy, the initial weight of each instant is set to one, giving all instants equal weights. For updating the classifier's weight, a different method of approximation of two error weights is used. All of these changes increased the effectiveness of OCBoost over previous findings.

As can be noted, not all ensemble methods can handle all sorts of concept drift, as can be seen in earlier boosting techniques. Similarly, the boosting strategies give the samples weights based on current prediction. Consequently, this could result in some data instances being underweighted, which will impact the classification's accuracy. For this purpose, this paper proposes a boosting-based ensemble method which is a modified version of the Boosting-like online Learning Ensemble (BOLE) [21] in which retain Previous State (PS) policy is used to improve consecutive results. As a result, in order to allocate weight for instances, the proposed method keeps an instance classification record of both the current as well as previous states.

III. MATERIAL AND METHODOLOGY

In this paper, we propose an enhanced version of the Boosting-like Online Learning Ensemble (BOLE) [22], which we have renamed as an Enhanced Boosting-like Online Ensemble that retains Previous State (EBOLE-PS). It is an online data stream classification method that employs the ensemble method, boosting, and drift detection techniques. Hoeffding trees as base learner are applied as expert classifiers for ensemble construction, and a drift detector is used to handle concept change to improve accuracy.

A. ENSEMBLE CONSTRUCTION

It has been observed that traditional data mining techniques have failed to process stream data, making the task difficult. In the case of data stream classification, a simple classification method degrades its performance due to variations in the target concepts. One method for addressing this issue is to develop an adaptive learning method using ensemble of classifiers, which employs multiple classifiers. In dealing with the large volume and concept variation problem in data streams, ensemble-based classifiers outperform single classification and are more accurate.

Boosting ensembles are designed as follows: The data instances (DI) are initially fed to the first base classifier with equal weights (DI*Wi) and the weights of instances are updated after each iteration by a policy in which incorrectly classified data instances (DIe) receive the highest weights for the next round (DIe*Wh). In contrast, correctly classified instances (DIc) will receive lower weights (DIc*Wl). This technique is beneficial for lowering false positive rates by increasing the selection probability of incorrectly classified instances to judge more times.

The proposed method builds an ensemble using 10 hoeffding trees as base learners and the same ensemble construction rule as the BOLE and ADOB methods. Before processing a new set of data instances, the base experts (h) are sorted ascending to achieve more accurate results. Simultaneously, the assumption is made that if a less accurate classifier can correctly predict results; there is no need to recheck results from more accurate classifiers.

The working of a proposed EBOLE-PS ensemble consists of three major parts:

A.1 CLASSIFIER VOTES

The weights of ensemble get calculated after each iteration. The technique of weight calculation is as follows:

i. Lamda (λ) is a Poisson distribution variable that is unique to each input data instance.

ii. ʎsc,m is the sum of the λ values for correctly classified examples by the base model at stage m and;

iii. ʎsw,m is the sum of incorrectly classified instances.

iv. If both these values (λsc,m and λsw,m) are non-zero, then error is calculated as follows using a simple formula as in (1):

$$\varepsilon m = (\lambda sw,m)/(\lambda sc,m + \lambda sw,m). \quad (1)$$

It means, Error ← (wrongly classified instances) / (Total Instances).This calculation is based on the current classification results.

If Error (Ɛm) < 50% then using ADOB classification method β_m is calculated as in (2).

$$\beta m = \frac{\varepsilon m}{(1-\varepsilon m)}. \quad (2)$$

In each iteration, the ensemble weight is calculated using formula in (3).

$$Lm\ (\dot{x}) = \dot{y}\log\left(\frac{1}{\beta m}\right). \quad (3)$$

Finally, all the prediction of classifiers is arranged in ascending order based on its weight and highest weight prediction is selected for the next phase.

```
Algorithm 1: Accuracy calculation modified
            version (train on instance)
Input: ensemble size N, instance I
Output: Expert arranged as per accuracy
1. accuracy[] ← N
2. for m ← 1 to N do
       a. λsw,m ← ( λsw,m + λoldsw,m )/ 2
       b. λsc,m ← (λsc,m + λoldsc,m )/ 2
       c. accuracy[m] ← λsc,m + λsw,m
3. if (accuracy[m] != 0.0)
       a. accuracy[m]←(λsc,m
          /accuracy[m])
4. End
```

A.2 BOOSTING STRATEGY

This section describes how the proposed method evolved from existing boosting methods. As previously stated, Adaboost is the foundation of online boosting algorithms that handle stream classification.

In any boosting algorithm, instances are weighted based on their prediction accuracy at each iteration. The most incorrectly predicted instances are given a higher weight. For instance voting, our proposed approach takes into account not only current predictions but also previous predictions. Assuming boldly that instances that have been repeatedly misclassified should be given more weight than those that have recently been misclassified. Algorithm 1 depicts the pseudo-code for the retain previous state (PS) policy for the java method getVotesForInstance, which is part of the boosting algorithms implemented in the MOA [23] framework.

Here $\lambda sc,m$ *and* $\lambda sw,m$ are used to indicate correctly and wrongly predicted instance votes respectively. These two variables are available in original method which keeps track of current instance prediction. We have added two extra parameters ($\lambda oldsw,m$ *and* $\lambda oldsc,m$) to this by taking previous state classification results into account. So that the instances that are repeatedly misclassified should be given higher weight. The proposed method uses an average of current and previous results to add recurrently incorrect predictions and thus increase the weight of such instances. The average weight of correct and incorrect classification is considered for each instance. As shown in the algorithm line number 2.a and 2.b part, the average of two continuous results is used to calculate the final weight of instances.

$$h(m) = \sum_{m=1}^{\infty} \left(\frac{\lambda sc,m}{\lambda sc,m + \lambda sw,m} \right). \qquad (4)$$

As shown in (4) weight of each classifier is calculated based on prediction accuracy.

A.3 DRIFT DETECTOR METHOD

The proposed method uses a different concept drift detection method based on error distance rather than a window-based method. The same method was used by the BOLE-based algorithm.

We decided that EBOLE-PS would use Drift Detection Method DDM [24] after reviewing the results of a recent comparison of concept drift detectors [25], which leads to the conclusion that the (DDM) is a good choice in all sorts of datasets. Algorithm 2 demonstrates the operation of DDM in simple steps.

```
Algorithm 2 : Drift detector Method
Input : Classifiers prediction (yes, no), Alert level, D-
        Danger level
Output : Flag(Alert, Danger)
  1.  For each input batch
      E  – Errors Distance (Average) ←1
      N – Incorrectly classified instances ←0
      Sq – Mean Square Error (Average) ← 0
      d – Standard deviation of error ←1
      i← i+1
  2.  If (classifier prediction is wrong)
              i.   N ← N+1
              ii.  E ← ((N-1)*E +i)/ N
              iii. Sq ← ((N-1 ) * Sq + $i^2$)/ N
              iv.  D ← Sqrt(Sq – $E^2$)
      End if
  3.  If (( $E_i$+$d_i$ ) greater than or equal to ($E_{min}$+ 2* $d_{min}$))
              Flag ← Alert
  4.  If (( $E_i$+$d_i$ ) greater than or equal to ($E_{min}$+ 3* $d_{min}$))
              Flag ← Danger
  5.  Return Flag
```

Thus, proposed EBOLE-PS works on instance selection strategy and allows more classifiers to vote. It also employs the error-based concept drift detection method DDM, resulting in a greater ensemble accuracy. Fig. 2 shows a flowchart of the system's overall work flow.

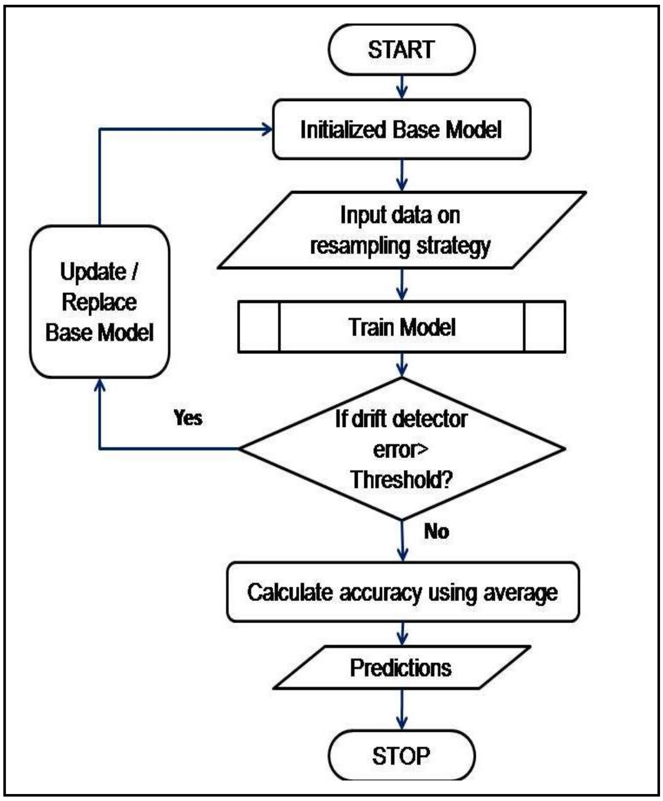
START
Initialized Base Model
Input data on resampling strategy
Train Model
If drift detector error> Threshold?
Yes
Update / Replace Base Model
No
Calculate accuracy using average
Predictions
STOP

Figure 2. Flowchart of overall working of EBOLE-PS

IV. DATASETS

This section described the datasets used in the comparative analysis of the performance of the existing methods and the proposed EBOLE-PS method. Four real-world datasets and four artificial datasets with unpredictability, noise, and a large amount of data are chosen. All of the datasets chosen are also freely available online, with the majority of them available on the MOA website [23].

A. REAL-WORLD DATASETS

The selected four Real datasets are with varying sizes and unknown drift.

Adult dataset with 15 attributes and 2 classes is highly imbalanced. It falls under the binary classification problem.

Electricity dataset contains data from the electricity market of Australian New South Wales, with 45,312 instances and eight attributes.

KDDCUP'99 dataset, which was collected for The Third International Knowledge Discovery and Data Mining Tools Competition, contains 42 different attributes and two classes.

Cover type stores data collected for a 30-meter square of forest from the US Forest Service (USFS) Region. It has over 500,000 instances with 54 attributes and many classes.

B. SYNTHETIC DATASETS

The 4 different synthetic datasets LED, Hyper plane, SEA, Sine with different drifts have been selected for the experiment.

LED: This data set, which has 24 parameters and 10 classes, is an example of gradual drift. For the experiment we used it with two different sets of it by random seed parameter as 1 and 4.

Hyper plane: This is a binary class dataset with d-dimensional space that constantly changes position and orientation. For the experiment, we created two versions of this dataset, one with 5% and one with 10% noise.

SEA: This dataset illustrates abrupt concept drift.

Sine: It is an example of a dataset with abrupt concept drift. All these synthetic data streams can be generated using MOA [23] framework.

IV. EXPERIMENTAL CONFIGURATION

This section describes the experiments that were carried out in order to test and evaluate the proposed EBOLE-PS algorithm. It is specifically compared to boosting-based ensembles OCBoost, OzaBoostAdwin, ADOB, and the base method BOLE to perform comparative analysis.

The Interleaved Test-Then-Train methodology is used to test instances before using them for training. This procedure ensures that each instance has passed through both phases. Because both ADOB and BOLE methods take less time and use less memory, the proposed modified method is only compared in terms of accuracy and kappa measure. All methods are based on boosting ensemble concepts; they have the same parameters, making comparison simple. We chose Hoeffding Tree as base classifiers with a number of experts set to ten.

The experiments are run on an Intel Core i5 processor with 4GB of main memory and Windows 10 64-bit. OZABOOST ADWIN and ADOB used the ADWIN drift detector to identify concept drifts. The formal parameter of ADWIN is, which indicates the maximum global error, and its default value is set to 0.002.

BOLE and EBOLE-PS, on the other hand, use Drift Detector Method DDM, as drift detector. All of these base methods and drift detector codes are available with the MOA framework.

The DDM parameters for the BOLE are as follows: n is the minimum number of processed instances before a drift can be detected which is set to 7, w is the standard deviation to raise warnings which is set to 1.2, and d is the out-control level which is set to 1.95. For our method, we set DDM to the same value as of based BOLE method (n = 7, w=1.2 and d=1.95)

V. RESULTS AND DISCUSSION

Every method has been run ten times to compute accuracy, execution time, and memory utilisation. For the final comparison, the average of all metrics is used.

A. ACCURACY ANALYSIS

Table 1 shows the accuracy of prediction achieved for OCBOOST, OZABOOST ADWIN, ADOB, BOLE, and EBOLE-PS. All are put through their paces on both artificial and real-world datasets. In the case of an LED dataset with one concept drift, EBOLE-PS performed the best (73.41 percent), followed by BOLE and OZABOOST. The ADOB method has degraded performance, whereas OCBOOST has performed the worst, with an average accuracy of 17.53 percent.

All methods' performance deteriorated with the addition of four more drifts in the LED dataset. However, EBOLE-PS performed well in this case as well, with only a 0.23 percent drop from a single drift LED data set. OCBOOST performance, on the other hand, is the worst in both cases.

The hyper plane dataset comes in two versions: one with 5% noise and one with 10% noise. In the presence of 5% noise, OCBOOST performed well, with the highest average accuracy of 89.5%, closely followed by EBOLE-PS, which is followed by BOLE, OZABOOST, and ADOB. In the case of Hyper (10% noise), as more noise is added to the data, the methods take a little longer to detect changes with slow recovery. OCBOOST's performance has been reduced by about 4%, but it still has a high accuracy. EBOLE-PS has a slightly lower accuracy of 0.08 percent than OCBOOST. These two methods have been followed by the BOLE, ADOB, and OZABOOST. Examples of gradual drift include Hyper plane and LED datasets. In a general analysis based on overall performance, the methods that performed well were: EBOLE-PS, BOLE, OZABOOST, ADOB, OCBOOST, with differences in LED dataset ranging from 0.43 percent to 55.79 percent, and differences in Hyper plane dataset ranging from 0.67 percent to 10.34 percent.

SEA and Sine datasets are with abrupt drift where average performance of all methods is above 80%. In the SEA dataset, EBOLE-PS had the highest accuracy (87.87%), followed by OCBOOST, BOLE, ADOB, and OZABOOST. The difference between high and low accuracy ranges from 0.93 percent to 4.23 percent. In comparison to other datasets, SINE dataset accuracy is greater than 97 percent using all tested methods. It may be due to symmetric nature of this dataset. OCBOOST has a score of 98.99 percent, while BOLE has a score of 97.84 percent.

The results in Table 1 show that the suggested EBOLE-PS approach has better performance accuracy than all of the compared approaches when the average of all results is considered. In the case of synthetic datasets, we employed hyper plane with 5% and 10% noise to see how noise affected ensemble performance. However, the results show that noise has no effect on the proposed method's performance. Similarly, in the case of the LED dataset, drifts have grown, although EBOLE-PS still outperforms other ensemble approaches.

To summarise the overall accuracy comparison, in the case of abrupt dataset like Sine, OCBOOST and EBOLE-PS perform better, followed by BOLE, ADOB, and OZABOOST. EBOLE-PS yielded the best results for gradual drifted datasets, but LED yielded different results than Hyper plane. In terms of LED, OCBOOST was the worst performer. ADOB has also demonstrated extremely low accuracy. In comparison, OCBOOST has provided excellent performance for hyper planes.

EBOLE-PS has fared better while analysing real-world data sets for the Adult and Covertype datasets. Due to its updating technique, EBOLE-PS had recorded higher accuracy in the Cover type and Adult dataset with 87.7% and 83.16% respectively.

The effectiveness of the proposed approach is demonstrated by the consistent results for a number of datasets, in situations with both gradual and abrupt concept drifts, as well as for real-world datasets. For example, the results for LED datasets (i=1 and i=4) varied dramatically, with differences ranging from 55.71 percent in the case of OCBOOST and EBOLE-PS to 17.43 percent in the case of ADOB and EBOLE-PS. The obtained accuracies for Covertype is high among the all compared algorithms. Furthermore, the 73.41 percent accuracy obtained in the LED (i=4) dataset is boldly stated to be the highest for this dataset in the case of all compared online ensemble methods. Besides that, EBOLE-PS is consistently performing to near accurate than the other two tested ensemble methods as well as the original methods ADOB and BOLE. Finally, the overall performance of EBOLE-PS is consistently high and comparable to that of BOLE.

To verify overall performance, each ensemble's average accuracy on all types of datasets is calculated. We can observe that the EBOLE-PS technique outperformed OCBOOST, OZABOOST ADWIN, ADOB, and BOLE by 14.45%, 4.79%, 12.67%, and 1.30% respectively.

We can state that the proposed system has produced good results for various types of drifts as well as data of various sizes. To make these findings more concrete, we consider another comparison parameter, the Kappa measure. The Kappa measure is used to determine how accurately a system operates by considering the correct and total results. The kappa formula is as shown in (5) and (6).

$$Ko = \frac{Correctly\ classified\ instances.}{Total\ instances}. \quad (5)$$

$$K_e = \left(\frac{+ve\ testing}{100} * \frac{+ve\ training}{100}\right) + \left(\frac{-ve\ testing}{100} * \frac{-ve\ training}{100}\right). \quad (6)$$

Table 2 displays the results of the kappa measure obtained by the proposed method as well as all boosting algorithms.

The performance of ensemble is also checked by increasing size of datasets. We used synthetic datasets of 100K, 200K, and 500K instances, respectively. On all the ensembles, Table 3 provides the average performance metric for different instance sizes. It is clear from the results that in the case of abrupt drift, such as Sine and SEA datasets, accuracy is higher for 200K and 500K instances, whereas in the case of gradual drift, such as Hyper plane and LED, prediction becomes more accurate as size increases.

Figs. 3, 4 and 5 compare the accuracy of ensembles for the Adult, Hyperplane, and SEA datasets. As demonstrated in the diagram, EBOLE-PS has obtained good accuracy from the start on the Adult dataset, and its performance has steadily improved following drift. Due to the gradual drift in the hyperplane dataset all other approaches degrade performance, whereas EBOLE-PS maintains its stability. On the SEA dataset, all algorithms showed fluctuation at first abrupt drift point, but only ADOB and EBOLE-PS were able to cope with the fluctuations and maintain their performance. The suggested system recognizes changes over time and adjusts to changing environments to handle the concept drift issue, as seen in all three graphs.

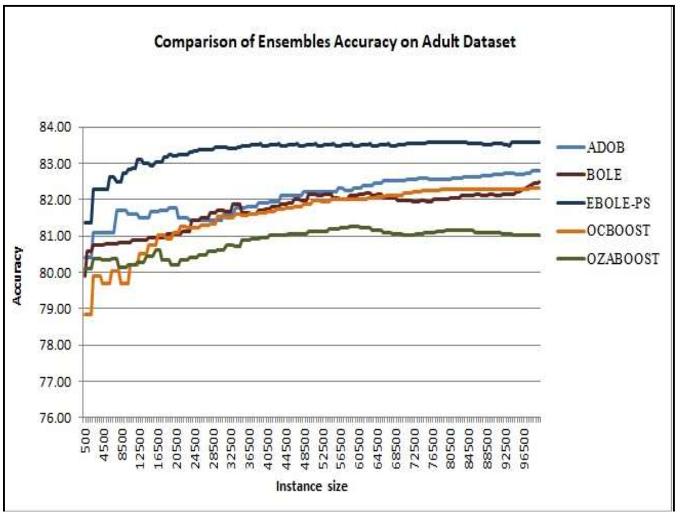

Figure 3 Comparison of classification accuracy on ADULT datasets

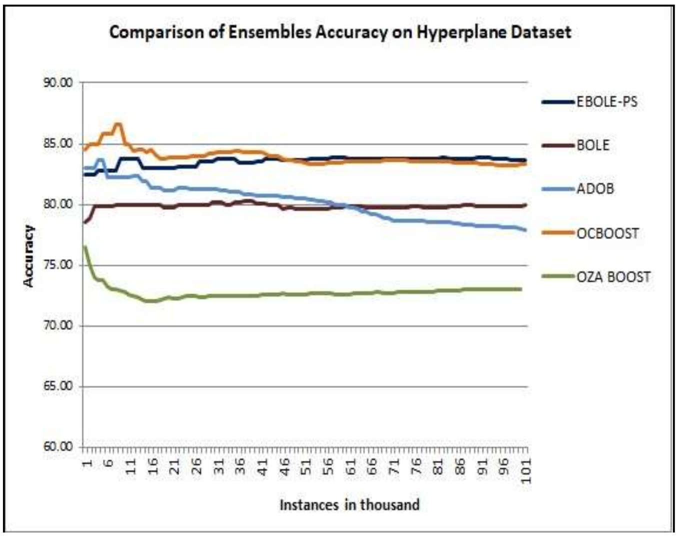

Figure 4 Comparison of classification accuracy on Hyper plane datasets

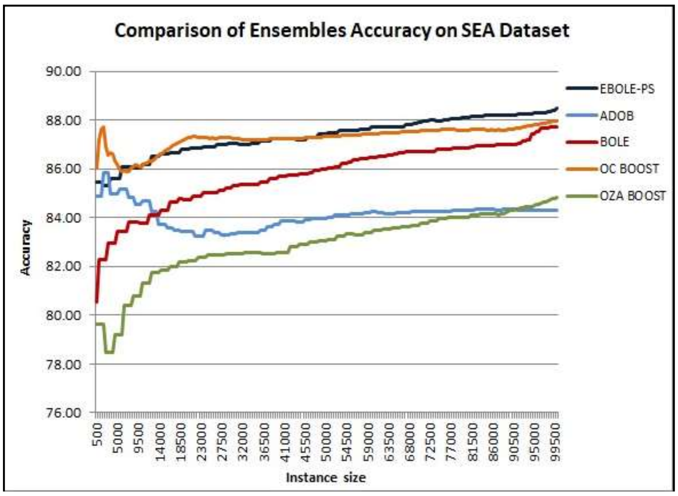

Figure 5 Comparison of classification accuracy on SEA datasets

Table 1. Average Accuracy in percentage on Real world and Synthetic Datasets for OCBOOST, OZABOOST ADWIN, ADOB, BOLE and proposed EBOLE-PS.

Datasets	OCBOOST	OZABOOST	ADOB	BOLE	EBOLE-PS
LED(i=4)	17.53	71.9	55.81	72.89	**73.24**
LED(i=1)	17.53	55.71	55.81	72.89	**73.41**
Hyper plane (5%)	**89.5**	79.68	78.04	84.68	88.36
Hyper plane (10%)	**83.66**	72.79	78.04	79.93	83.58
SEA	86.94	83.64	84.23	86.33	**87.87**
Sine	**98.39**	98.33	98.34	97.84	98.31
Covertype	59.7	85.41	31.87	87.68	**87.7**
Adult	81.47	80.82	82.11	81.82	**83.16**
KDDCup99	92.99	**96.04**	95.46	94.14	95.81
Electricity	90.77	90.71	76.56	**91.75**	91.52
Average	**71.85**	**81.50**	**73.63**	**85.00**	**86.30**

Table 2. Accuracy in terms of Kappa measure percentage on Real worlds and Synthetic Datasets for OCBOOST, OZABOOOST, ADOB, BOLE and proposed EBOLE-PS.

Datasets	OCBOOST	OZABOOST	ADOB	BOLE	EBOLE-PS
LED(i=4)	8.38	68.77	50.9	69.88	70.26
LED(i=1)	79.01	59.36	56.08	59.86	67.16
Hyper plane (5%)	70.95	64.01	63.67	69.7	72.99
Hyper plane (10%)	96.75	96.64	96.65	95.65	96.59
SEA	32.75	78.43	15.23	81.65	81.67
Sine	46.58	46.77	51.5	51.89	54.39
Cover type	85.72	91.91	90.74	88.04	91.52
Adult	81.06	80.92	48.8	83.11	82.64

Table 3. Accuracy in percentage on Synthetic Datasets with 100K, 200K and 500K instances for ADOB, BOLE and proposed EBOLE-PS

Datasets	Instance size	ADOB	BOLE	EBOLE-PS
LED	100K	55.81	72.89	**73.24**
	200K	54.31	73.81	**73.87**
Hyper plane	100K	79.72	84.68	**88.36**
	200K	79.72	85.17	**88.39**
	500K	63.01	86.07	**88.66**
SEA	100K	84.23	86.33	**87.87**
	200K	85.62	88.39	**89.11**
	500K	85.48	88.14	**88.96**
Sine	100K	98.34	97.84	98.31
	200K	98.82	98.47	**98.82**
	500K	99.26	99.06	**99.29**

VI. CONCLUSIONS

Data stream classification is one of the most widely used mining techniques for various online data analyses today. The boosting strategy is always given better results when it comes to speeding up the mining process. The proposed method improved classification process accuracy in boosting base ensemble classifiers. This paper proposes preserving "results of previous stage" strategies in order to improve the accuracy of online boosting methods, especially in different concept drifts scenarios. More specifically, we investigated the effects of retaining previous data instance classification records for the next phase so that only the current prediction does not receive more weight. This phenomenon is used to keep track of incorrectly classified instances so that they can be tested more frequently, resulting in better outcomes due to more rectification. The results show that the proposed changes to the weighting strategy of boosting method improved prediction accuracy.

The effect of drift detector`s parameterization on final accuracy over different types of datasets could also be investigated, and thus the system's current performance could be improved. Further to that, we have not thoroughly examined the impact of changing the drift detectors' warning level and instance limit, which may have an effect on the drift detection percentages.

Current ensembles function with data streams generated by stream generators, but real-world data may have difficulties like as imbalanced classes, missing values, outliers, and so on, necessitating the development of a full-proof framework that addresses data pre-processing.

To summarize the above, the proposed innovative approach EBOLE-PS, can be viewed as an experimental reorganization of the boosting concept algorithmic solution, which may lead to a demonstrably better ensemble technique in the future.

Acknowledgements

The authors are thankful for Bharati Vidyapeeth deemed to be university, College of Engineering Pune for its support and valuable guidance.

References

[1] H. M. Gomes, J. P. Barddal, A. F. Enembreck, and A. Bifet, "A survey on ensemble learning for data stream classification," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1-36, 2017. https://doi.org/10.1145/3054925.

[2] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Inf. Fusion*, vol. 37, pp. 132–156, 2017. https://doi.org/10.1016/j.inffus.2017.02.004.

[3] T. Phanomsophon, N. Jaisue, N. Tawinteung, L. Khurnpoon, and P. Sirisomboon, "Classification of N, P, and K concentrations in durian (Durio Zibethinus Murray CV. Mon Thong) leaves using near-infrared spectroscopy," *Eng. Appl. Sci. Res.*, vol. 49, no. 1, pp. 127–132, 2022. https://doi.org/10.14456/easr.2022.15.

[4] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2049 LNAI, no. May, pp. 249–257, 2001, https://doi.org/10.1007/3-540-44673-7_12.

[5] H. Abdulsalam, D. B. Skillicorn, and P. Martin, "Classification using streaming random forests," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 1, pp. 22–36, 2011, https://doi.org/10.1109/TKDE.2010.36.

[6] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with drifting streaming data," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 1, pp. 27–39, 2014, https://doi.org/10.1109/TNNLS.2012.2236570.

[7] K. Nishida and K. Yamauchi, "Adaptive classifiers-ensemble system for tracking concept drift," *Proc. of the Sixth Int. Conf. Mach. Learn. Cybern. ICMLC 2007*, vol. 6, no. August, pp. 3607–3612, 2007, https://doi.org/10.1109/ICMLC.2007.4370772.

[8] J. Liu, G. S. Xu, S. H. Zheng, D. Xiao, and L. Z. Gu, "Data streams classification with ensemble model based on decision-feedback," *J. China Univ. Posts Telecommun.*, vol. 21, no. 1, pp. 79–85, 2014, https://doi.org/10.1016/S1005-8885(14)60272-7.

[9] L. L. Minku and X. Yao, "DDD: A new ensemble approach for dealing with concept drift," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 4, pp. 619–633, 2012, https://doi.org/10.1109/TKDE.2011.58.

[10] H. He and S. Chen, "Towards incremental learning of nonstationary imbalanced data stream: A multiple selectively recursive approach," *Evol. Syst.*, vol. 2, no. 1, pp. 35–50, 2011, https://doi.org/10.1007/s12530-010-9021-y.

[11] K. K. Wankhade, K. C. Jondhale, and S. S. Dongre, "A clustering and ensemble based classifier for data stream classification," *Appl. Soft Comput.*, vol. 102, p. 107076, 2021, https://doi.org/10.1016/j.asoc.2020.107076.

[12] P. Zhang, X. Zhu, J. Tan, and L. Guo, "Classifier and cluster ensembles for mining concept drifting data streams," *Proc. of the IEEE Int. Conf. Data Mining, ICDM*, pp. 1175–1180, 2010, https://doi.org/10.1109/ICDM.2010.125.

[13] N. C. Oza and S. Russell, "Experimental comparisons of online and batch versions of bagging and boosting," *Proc. of the Seventh ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 359–364, 2001, https://doi.org/10.1145/502512.502565.

[14] J. Gama, P. P. Rodrigues, and R. Sebastião, "Evaluating algorithms that learn from data streams," *Proc. of the ACM Symp. Appl. Comput.*, pp. 1496–1500, 2009, https://doi.org/10.1145/1529282.1529616.

[15] Okfalisa *et al.*, "Forecasting company financial distress: C4.5 and adaboost adoption," *Eng. Appl. Sci. Res.*, vol. 49, no. 3, pp. 300–307, 2022, https://doi.org/10.14456/easr.2022.31.

[16] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Trans. Neural Networks*, vol. 22, no. 10, pp. 1517–1531, 2011, https://doi.org/10.1109/TNN.2011.2160459.

[17] R. C. Samant and D. D. M. Thakore, "A rigorous review on an ensemble based data stream drift classification methods," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 5, pp. 380–385, 2019. https://doi.org/10.26438/ijcse/v7i5.380385.

[18] R. C. Samant and S. H. Patil, "Adequacy of effectual ensemble classification approach to detect drift in data streams," *Proceedings of the 2022 International Conference for Advancement in Technology (ICONAT)*, Jan. 2022, pp. 1–6. https://doi.org/10.1109/ICONAT53423.2022.9725854.

[19] S. R. Nikunj oza, "Online bagging and boosting," in *8th Int. Workshop on Artificial Intelligence and Statistics*, 2001, pp. 105–112.

[20] S. G. T. D. C. Santos, P. M. Gonçalves, G. D. D. S. Silva, and R. S. M. De Barros, "Speeding up recovery from concept drifts," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8726 LNAI, no. PART 3, pp. 179–194, 2014, https://doi.org/10.1007/978-3-662-44845-8_12.

[21] R. S. M. d. Barros, S. Garrido T. de Carvalho Santos and P. M. Gonçalves Júnior, "A Boosting-like Online Learning Ensemble," *Proceedings of the* 2016 International Joint Conference *on Neural Networks (IJCNN)*, 2016, pp. 1871-1878, https://doi.org/10.1109/IJCNN.2016.7727427.

[22] R. Pelossof, M. Jones, I. Vovsha, and C. Rudin, "Online coordinate boosting," *Proceedings of the 2009 IEEE 12th Int. Conf. Comput. Vis. Work. ICCV Work. 2009*, pp. 1354–1361, 2009, https://doi.org/10.1109/ICCVW.2009.5457454.

[23] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive Online Analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 1601–1604, 2010.

[24] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3171, no. September, pp. 286–295, 2004, https://doi.org/10.1007/978-3-540-28645-5_29.

[25] R. Samant and S. Patil, "Comparative analysis of drift detection techniques used in ensemble classification approach," *Proceedings of the International Conference on Recent Challenges in Engineering Science and Technology (ICRCEST 2K21)*, 2021, pp. 201–204.

Mrs. RUCHA CHETAN SAMANT is a Research Scholar (Computer Engineering) at Bharati Vidhyapeeth Deemed to be university, Pune. She has completed her B. E computer and M. E. computer from Mumbai University. She is currently working as Assistant Professor in Gokhale Education Society`s R. H. Sapat College of Engineering, Management Stuies and Research, Nashik. She has total 19 years of teaching experience with more than 16 research article published in national and international conferences and journals.

Dr. SUHAS PATIL, is currently working as a Professor of Computer Department at Bharati Vidhyapeeth Deemed to be university, Pune. He has completed his B. E computer from WIT, Solapur and M. E. computer from GCOE Pune. He has persuaded his Ph. D. (Computer Engineering) from BVDU COE, Pune. He has more than 31 years of teaching experience and guided more than 18 PhD students and 100 M. E. Students. He has published more than 350 research papers in international journals (Google scholars, Web of Science and Scopus indexed) and around 100 papers in international conferences. He was Member, Academic Council, Member, Faculty of Engineering and Technology, and Chairman BOS (Computer and IT) Bharati Vidyapeeth Deemed University, Pune. He is also Is Member for Board of Study (Computer Engineering), BVDU, Pune, Member BOS (comp), Cummins COE, Pune, (Autonomous Institute) Member BOS (comp) VIIT, Pune (Autonomous Institute), Member BOS IT JSPM, Pune (Autonomous Institute).