Smart pools of data with ensembles for adaptive learning in dynamic data streams with class imbalance

Radhika Vikas Kulkarni[1], S. Revathy[1], Suhas Haribhau Patil[2]
[1]Department of Computer Science Engineering, Sathyabama Institute of Science and Technology, Chennai, India.
[2]Department of Computer Science Engineering, Bharati Vidyapeeth's College of Engineering, Pune, India

Article Info

Article history:

Received Jun 3, 2021
Revised Dec 16, 2021
Accepted Dec 28, 2021

Keywords:

Adaptive learning
Class imbalance
Concept drift
Data streams classification
Ensemble
Online learning

ABSTRACT

Streaming data incorporates dynamicity due to a nonstationary environment where data samples may endure class imbalance and change in data distribution over the period causing concept drifts. In real-life applications learning in dynamic data streams, is vitally important and challenging. A combined solution to adapt to class imbalance and concept drifts in dynamic data streams is rarely addressed by researchers. With this motivation, the current communication presents the online ensemble model smart pools of data with ensembles for class imbalance adaptive learning (SPECIAL) to learn in skewed and drifting data streams. It employs an ageing-based G-mean maximization strategy to adapt to dynamicity in data streams. It employs smart data-pools with the local expertise ensemble to classify samples lying in the same data-pool. The empirical and statistical study on different evaluation metrics exhibits that SPECIAL is more adaptive to class imbalanced dynamic data streams than the state-of-the-art algorithms.

Corresponding Author:

Radhika Vikas Kulkarni
Department of Computer Science Engineering, Sathyabama Institute of Science and Technology
Chennai, India
Email: radhikavikaskulkarni@gmail.com

1. INTRODUCTION

The current era of information technology demands predictive models for numerous real-life applications like social media sentiments extraction [1], financial risk prediction [2], network intrusion identification [3], and so on. Such applications produce a continuous stream of endless data with high volume and speed [4]-[6]. Data streams possess dynamicity due to the varying distribution of data in them. The change in data distribution is referred to as "concept drift" [5], [7]. The unequal number of class instances of a data stream cause skewness [8]-[9]. Learning in nonstationary data streams with the class imbalance and drifting concepts is a challenging task in computational intelligence [10].

Learning in nonstationary data streams has attracted many researchers [7]. Ensemble learning employs a combination of single classifiers to augment the generalization capability and is more suitable to handle dynamicity in data streams [4], [5], [11]-[13]. Block-based ensembles like accuracy weighted ensemble (AWE) [14], accuracy updated ensemble (AUE) [15], dynamic weighted majority for imbalance learning (DWMIL) [16], adaptive chunk-based dynamic weighted majority (ACDWM) [17] tackle to concept drifts by inserting a new learner trained on a new block of data. Online ensembles like ADWIN Bagging (BagADWIN) [18], online accuracy updated ensemble (OAUE) [19] train the model on each incoming data sample avoiding waiting for a block or whole training dataset. The implicit adaptation to concept drifts in data streams leads to passive learning algorithms like dynamic weighted majority (DWM) [20], anticipative dynamic adaptation to concept change (ADACC) [21], dynamic classifier selection (DCS) [22]. However, explicit drift detection leads to an active learning approach as narrated in [23]-[26].

A rich literature on the class imbalance problem is available [8], [27] that describes three broad approaches to handle skewness in data: i) sampling-based, ii) algorithm-based, and iii) cost-sensitive. Most of the researchers have handled the problem of class imbalance and dynamicity in data streams separately. The combined solution to both problems is presented by comparatively fewer researchers [28]-[29].

Through this correspondence, we provide a novel combined solution to the issue of class imbalance and concept drifts in dynamic data streams. We propose an online ensemble classifier smart pools of data with ensembles for class imbalance adaptive learning (SPECIAL) to classify binary dynamic data streams. The contributions of the proposed work are:

- It provides a passive drift detection model using an online ensemble with a test-then-train approach.
- It employs an ageing-based strategy to adapt to the dynamicity in data.
- It deals with the class imbalance in streaming data with the objective of G-mean maximization.
- It investigates empirical and statistical test results and compares the performance of the proposed algorithm SPECIAL with state-of-the-art ensembles on different performance measures using various real and synthetic benchmark datasets.

2. RESEARCH METHOD

The current section describes the problem of this research and the proposed methodology to solve it. The proposed methodology elaborates the training and testing phase of the proposed algorithm SPECIAL. The section further presents the experimental framework to test the performance of the algorithm SPECIAL.

2.1. Problem formulation

The data stream DS arrives as a sequence of massive data samples $\{d_1, d_2, d_3, \ldots\}$ where d_t is a data sample in the m-dimensional feature space arrived at time step $t = 1, 2, \ldots$. Associated with each of the data sample d_t there is a class label $c_t \in C=\{c_1, c_2, ..., c_L\}$ where L is the number of class labels. As our work focuses on a binary classification problem, we consider C=$\{0,1\}$ where '*0*' defines a negative class, and '*1*' defines a positive class. Let DS^0 be the set of negative data samples and DS^1 be the set of positive data samples. We consider class imbalance in the data stream where $|DS^0| >> |DS^1|$. A concept in the data stream is defined by a joint distribution $P_t(DS, C)$. At time step t, it generates a tuple (d_t, c_t). The dynamicity in the data stream may cause concept drift where its joint distribution changes over the period i.e. $P_{t-1}(DS, C) \neq P_t(DS, C)$ [7]. As an online learner progresses with the recently received data samples, it is suitable for handling the sequential incoming flow of data samples of the data streams [7], [30]. Hence, for the prediction of a class c_t of an input data sample d_t of a dynamic data stream, we propose an online classifier $f: DS \rightarrow C$.

2.2. Proposed methodology

Through this communication, we present an online adaptive ensemble SPECIAL. It is an ensemble of ensembles $E=\{e_1, e_2, ..., e_s\}$ where s is the number of sub ensembles to deal with dynamically drifting imbalanced data streams. We focus on the local expertise of each of the sub ensembles on the area of the m-dimensional feature space where the recent data sample has appeared. Thus, it prefers the predictions by the sub ensemble which exhibited better classification accuracy in this area.

The data samples in the same area of the m-dimensional feature space are mapped to a single data-pool which represents the m-dimensional hypersphere. The data samples in a data-pool point to their expertise sub ensemble e_i where $i \in \{1, 2, ..., s\}$ which gives the lowest classification error for the data samples in that data-pool. Each data-pool p_j, $j \in \{1, 2, ..., z\}$ where z is the number of existing data-pools is characterized by the following metadata at time step t:

a. A pool-master (PM_t^j): It is a representative data sample of the jth data-pool based on the minimum prediction error in its classification. At time step t, a tie with the same prediction errors, if any, is resolved by selecting the most recently mapped data sample to that pool as its pool-master.

b. Pool size (PS_t^j): It is the number of data samples mapped to the jth data-pool by the time step t.

c. Average prediction error (APE_t^j): It is the average prediction error incurred in the classification of all data samples in the jth data-pool by the time step t.

d. Prediction factor (PF_t^j): It is given as $(PF_t^j) = 1 - (APE_t^j)$. The higher value of (PF_t^j) indicates that the expertise sub ensemble associated with the jth data-pool shows better classification results.

e. Representative factor (RF_t^j): It is a ratio of pool size of the jth data-pool to the total number of data samples received by the time step t.

f. Weight factor (WF_t^j): It is a ratio of prediction factor (PF_t^j) to representative factor (RF_t^j) of the jth data-pool by the time step *t*. The pool with a higher weight factor is preferred as it gives better prediction with a lesser number of data samples mapped to it. And, as it has a smaller pool-size we can map more suitable data samples to it.

The proposed SPECIAL classifier employs a test-then-train approach. Through this approach, it first predicts a class of each sample (d_t, c_t) received at time step *t* and then uses the same for the training of the model. The detailed description of the testing phase and training phase of the learning model is given below.

2.2.1. Model testing phase

For each incoming data sample at time *t*, the proposed algorithm identifies its *K* nearest pools. It assigns the highest priority to one of these *K* nearest pool-masters if it is the most recently defined pool-master and has the highest weight factor (WF_t^j). Then the expertise learner associated with the selected highest priority *jth* data-pool classifies the newly arrived data sample. On correct classification, the new data sample is mapped to the selected data-pool, and metadata of that data-pool is updated.

At the start when no data-pool is created and no pool-master is available, the first incoming data sample itself forms a new data-pool and becomes a pool-master. It is tested on each sub ensemble $e_i \in E$ where $i \in \{1, 2, ..., s\}$. This newly formed data-pool points to the sub ensemble with the least prediction error. Accordingly, its metadata is updated. The prediction results are empirically and statistically examined on various evaluation metrics. The performance of the SPECIAL algorithm is compared with other state-of-the-art algorithms. Algorithm 1 describes the testing phase of the SPECIAL model.

Algorithm 1: SPECIAL model testing

```
Input: ($d_t$, $c_t$) is an incoming data instance at time $t=\{1, 2, …\}$ of data stream $DS$; $E=\{e_1, e_2, …, e_s\}$is an ensemble of $s$ ensembles; $P=\{ p_1, p_2, …, p_z\}$is a collection of existing $z$ data-pools.
Output: $\hat{c}_t$: A predicted class of data instance $d_t$.
P_near=Search_K-nearest-pools ($d_t$, K, P);
if (P_near==$\Phi$) {
   $\hat{c}_t$=Get_Prediction ($d_t$, E);
   $p_1$=Create_New_Data-pool ($d_t$); Update_Metadata ($p_1$);}
else {
   p=Select_Highest-prority_Pool (P_near);
   xe=Get_Associated_Expertise_Subensemble(p);
   $\hat{c}_t$=Get_Prediction ($d_t$, xe);
   if($c_t == \hat{c}_t$) {
      Map_to_Data-pool ($d_t$, p); Update_Metadata(p);}}
Evaluate the performance of the SPECIAL model on various evaluation metrics using ($\hat{c}_t$);
```

2.2.2. Model training phase

The proposed SPECIAL classifier employs online learning as presented by Oza N. [31]. Generally, the bootstrapping samples follow a normal distribution. Due to the unavailability of all training instances at the start and a huge volume of incoming streaming data, the normal distribution of bootstrapping samples is approximated by a Poisson (1) distribution, if the number of training instances $n \rightarrow \infty$.

The SPECIAL algorithm adapts to dynamicity in the data stream by assigning more weightage to the recently arrived data instances. It incorporates two predefined ageing metrics: 1) data-ageing metric β, $(0<\beta<1)$ and 2) sensitivity-ageing metric γ, $(0<\gamma<1)$ to give more weightage to the latest data and sensitivity (i.e. true positive rate). The function $f(c_t, c_L)$ returns 1 if the class label c_t of incoming data instance is c_L and 0 otherwise, where $L=\{0,1\}$. The function $g(c_t, \hat{c}_t)$ returns 1 if the class label of incoming data instance is correctly predicted and 0 otherwise. Let (DS_t^L) where $L=\{0,1\}$ be the metrics defining the percentages of negative class (if $L=0$) or positive class (if $L=1$) by time step *t*. Let (Se_t) be the metric defining sensitivity by time step *t*. The ageing-based computations of (DS_t^L) where $L=\{0,1\}$and (Se_t) are given by (1) and (2) respectively. Referring to these ageing-based computations shown in (1) and (2) it can be noticed that the ageing metrics β and γ force the old samples to contribute less to computations of metrics of class percentage and sensitivity, respectively.

$$DS_t^L = \beta \cdot DS_{t-1}^L + (1-\beta) \cdot f(c_t, c_L);\text{ where } L=\{0,1\} \tag{1}$$

$$Se_t = \gamma \cdot Se_{t-1} + (1-\gamma) \cdot g(c_t, \hat{c}_t) \tag{2}$$

When the difference between class percentages $|(DS_t^0 - DS_t^1)|$ crosses the predefined threshold Ψ, the proposed SPECIAL algorithm considers it as a scenario of class imbalance at time step t. To handle a class imbalance in binary data streams, it focuses on G-mean improvisation as described in [32]. It focuses on both positive (minority) and negative (majority) classes and deemphasises negative class only when the class imbalance results in a poor true positive rate. Accordingly, when data size $n \to \infty$, and there is a class imbalance in an incoming data stream, the number of copies B of the positive and negative class samples by Poisson (λ) approximated bootstrapping at time step t is given by (3). In the training phase, the sub ensembles associated with data-pools are trained by using a B number of copies of the data sample (d_t, c_t). The trained model of the SPECIAL classifier at time step t is used to test the unseen incoming data sample (d_{t+1}, c_{t+1}) at the next time step $t+1$. Algorithm 2 describes the training phase of the SPECIAL model.

$$B \sim \begin{cases} Poisson(Se_t); & if\ |(DS_t^0 - DS_t^1)| > \Psi\ and\ c_t = 0 \\ Poisson\left(\dfrac{DS_t^0}{DS_t^1}\right); & if\ |(DS_t^0 - DS_t^1)| > \Psi\ and\ c_t = 1 \\ Poisson(1); & if\ |(DS_t^0 - DS_t^1)| \leq \Psi \end{cases} \tag{3}$$

Algorithm 2: SPECIAL Model Training

```
Input: ($d_t$, $c_t$) is an incoming data instance at time $t=\{1, 2, \ldots\}$ of data stream $DS$;
$E=\{e_1, e_2, \ldots, e_s\}$ is an ensemble of $s$ ensembles; $P=\{p_1, p_2, \ldots, p_z\}$is a collection of $z$
data-pools; $\hat{c_t}$ is a predicted class of instance $d_t$ in the testing phase; $\Psi$ is the
imbalance threshold.
Output: An updated model of SPECIAL.
Initialize: $(DS_t^0)$=$(DS_t^1)$=0; $(Se_t)$=0;
if $(c_t \neq \hat{c}_t)$ {
  $p$=Select_Highest-prority_Pool($P$);
  $xe$=Get_Associated_Expertise_Subensemble($p$);
  $\hat{y_t}$=Get_Prediction ($d_t$, $xe$);
  if $(c_t == \hat{y}_t)$ {
     Map_to_Data-pool ($d_t$, $p$); Update_Metadata($p$);
  else {$p\_new$ =Create_New_Data-pool ($d_t$); Update_Metadata ($p\_new$); $P=P \cup p\_new$;}}
Update metrics $(DS_t^0)$, $(DS_t^1)$ using equation (1) and $(Se_t)$ using equation (2);
if $(|(DS_t^0 - DS_t^1)| > \Psi\ and\ c_t == 0)$ {
  $B \sim Poisson(Se_t)$; Repeat $B$ times training of the ensemble $E=\{e_1, e_2, \ldots, e_s\}$; }
else if $(|(DS_t^0 - DS_t^1)| > \Psi\ and\ c_t == 1)$ {
  $B \sim Poisson\left(\frac{DS_t^0}{DS_t^1}\right)$; Repeat $B$ times training of the ensemble $E=\{e_1, e_2, \ldots, e_s\}$; }
else { $B \sim Poisson(1)$; Repeat $B$ times training of the ensemble $E=\{e_1, e_2, \ldots, e_s\}$; }
```

2.3. Experimental framework

This section describes the experimentation framework used to assess the performance of SPECIAL. It provides the details of various datasets used for the experimentation. It also specifies the performance metrics used and the necessary experimental setup for the current study.

2.3.1. Datasets

A variety of datasets are widely used in the study on dynamic data streams as mentioned in [33]. The current study uses six popular binary datasets. Table 1 summarizes the description of these datasets. The used benchmark data streams have variations in the number of samples, the number of attributes, imbalance ratio, and types of drifts. Electricity pricing, agrawal and rotating hyperplane (RotHyperplane) datasets are taken from massive online analysis (MOA) framework [34]. The datasets like weather, SEA and rotating checker board with constant drift rate (RotChBoard-CDR) are available in the repository by [35].

2.3.2. Performance metrics

The presented research focuses on the classification of dynamic streams which may possess concept drift and skewness. Let *TP*, *FP*, *TN*, and *FN* be the number of true positive, false positive, true negative, and false negative data samples resulting in the binary classification, respectively. For skewed data, the only accuracy $((TP + TN)/(TP + TN + FP + FN))$ cannot justify the performance of the classifier as it gets influenced by the majority samples [7]-[8]. However, G-mean $\left(\sqrt{(TP/(TP + FN)) \cdot (TN/(TN + FP))}\right)$ focuses on the classification of both positive and negative samples as it gives a geometric mean of sensitivity and specificity. Hence, in addition to accuracy, the current study investigates the predictive capabilities of the SPECIAL using the metrics like G-mean, and F1-measure $\left(\frac{2 \cdot (TP/TP+FN) \cdot (TP/TP+FP)}{(TP/TP+FN)+(TP/TP+FP)}\right)$.

Table 1. The summary of datasets used

Dataset	Samples	Classes	Attributes	Positive Samples	Negative Samples	Type of Drifts
Weather (R)	18159	2	8	5698	12461	N/A
Electricity Pricing (R)	45312	2	8	19237	26075	N/A
SEA (S)	50000	2	3	18581	31419	Real, Abrupt
Agrawal (S)	100000	2	9	32656	67344	N/A
RotHyperplane (S)	200000	2	10	99935	100065	Real, Gradual
RotChBoard-CDR(S)	409600	2	2	204758	204842	Real, Gradual

(R)-Real, (S)-Synthetic, N/A-It has a drift, but its type is not available.

2.3.3. Experimental setup

The proposed SPECIAL algorithm builds an ensemble of ensembles. For the same, it blends five seminal ensembles as its base learners. The list of these sub ensembles used to build a learning model of SPECIAL is:

a. Hoeffding tree (HT) [36]: It is an incremental decision tree to learn in a streaming environment.
b. Dynamic weighted majority (DWM) [20]: It is a chunk-based ensemble that continuously evaluates the weights of its base learners considering the prediction results. It replaces the poor-performing base learner and updates the ensemble.
c. ADWIN bagging (BagADWIN) [23]: It is an extension of the online bagging algorithm [31] by integrating the ADWIN (ADaptive sliding WINdowing) algorithm [37] in it.
d. ADWIN boosting (BoostADWIN) [23]: It is an extension of the online boosting algorithm [31] by combining the ADWIN (ADaptive sliding WINdowing) algorithm [37].
e. Anticipative dynamic adaptation to concept change (ADACC) [21]: It is an incremental ensemble capable of handling recurring concept drifts using Kappa statistics.

The performance of SPECIAL is compared with seven state-of-the-art algorithms used in data stream classification. It is compared with its five sub ensembles as mentioned in the above list. In addition to that it is also compared with the following two classifiers:

a. Hierarchical linear four rates (HLFR) [26]: It is an online method for concept drift detection in the dynamic data stream.
b. Adaptive chunk-based dynamic weighted majority (ACDWM) [17]: It is a block-based ensemble method for concept drift detection in the streaming environment.

All algorithms are evaluated using the test-then-train approach. We define both data-ageing metric β and sensitivity-ageing metric γ as 0.9. Also, the threshold Ψ used for the detection of class imbalance scenario is set to 0.6. In the testing phase, we search for K=20 nearest data-pools.

3. RESULTS AND DISCUSSION

This part evaluates the performance of the proposed SPECIAL algorithm on a variety of datasets. It presents the empirical results of the experimentation. It also describes the statistical analysis of the current work.

3.1. Empirical results

We empirically test the performance of SPECIAL on three evaluation metrics: i) accuracy, ii) G-mean, and iii) F1-measure and compare it with seven state-of-the-art algorithms used to learn in the streaming environment. Tables 2 to 4 present the experimental results of all algorithms on three metrics. All results are given in percentages and the values in the parenthesis indicate the rank of the algorithm when tested on a specific dataset. The minimum value of average rank indicates the best performance of the algorithm.

Table 2 presents the accuracy results of all algorithms. SPECIAL gives the highest accuracy on real datasets weather and electricity. With the least value of the average rank, it shows the overall best performance on accuracy metric. The G-mean results of all algorithms are given in Table 3. Considering the overall average rank value on all datasets, SPECIAL provides the best G-mean results. Table 4 summarizes the F1-measure results of all algorithms. SPECIAL is the best performer with the minimum average ranking on F1-measure. Figure 1 depicts the overall average of ranks of all algorithms. It is noticed that the proposed algorithm SPECIAL with the least value of the overall mean of ranks beats all other state-of-the-art classifiers. The ageing factors used in SPECIAL gives more significance to the recent data that helps to adapt to the latest changes in the incoming data. SPECIAL incorporates the G-mean improvisation strategy with the Poisson (λ) approximated bootstrapping that focuses on the recalls of both positive and negative classes. Also, the employment of the ensemble of locally expertise sub ensembles in SPECIAL alleviates the limitations of an individual classifier and gives improved classification results. Thus, the SPECIAL provides better results for adaptive online learning in skewed dynamic data streams than other state-of-the-art learners.

Table 2. Accuracy percentage and ranking of all algorithms on all datasets

Dataset	Weather	Electricity	SEA	Agrawal	RotHyperplane	RotChBoard-CDR	Avg. Rank
HT	73.43(5)	80.33 (5)	94.1 (2)	85.32 (5)	84.09 (4)	60.21 (8)	(4.83)
DWM	70.13(7)	78.98 (6)	87.55(5)	87.87 (2)	89.86 (1)	71.95 (6)	(4.5)
OzaBagAdwin	75.01(2)	83.67 (4)	94.58(1)	87.01 (3)	88.01 (2)	85.37 (3)	(2.5)
OzaBoostAdwin	74.39(3)	88.13 (3)	88.17(4)	83.45 (7)	76.57 (7)	94.34 (1)	(4.17)
ADACC	73.57(4)	89.66 (2)	82.83(7)	84.78 (6)	82.77 (6)	80.03 (5)	(5)
HLFR	61.61(8)	60.82 (8)	61.44(8)	60.17 (8)	55.68 (8)	66.57 (7)	(7.83)
ACDWM	71.09(6)	76.48 (7)	87.21(6)	94.35 (1)	83.28 (5)	81.64 (4)	(4.83)
SPECIAL	75.08(1)	90.26 (1)	93.03(3)	86.05 (4)	86.31 (3)	93.1 (2)	(2.33)

Table 3. G-mean percentage and ranking of all algorithms on all datasets

Dataset	Weather	Electricity	SEA	Agrawal	RotHyperplane	RotChBoard-CDR	Avg. Rank
HT	63.16(6)	80.03 (5)	93.51(2)	82.66 (5)	84.08 (4)	60 (8)	(5)
DWM	70.45(2)	77.16 (6)	79.03(6)	85.09 (2)	89.86 (1)	71.95 (6)	(3.83)
OzaBagAdwin	60.66(7)	82.45 (4)	94.3 (1)	84.54 (3)	88.01 (2)	85.37 (3)	(3.33)
OzaBoostAdwin	67.85(4)	87.73 (3)	85.32(5)	81.16 (7)	76.57 (7)	94.34 (1)	(4.5)
ADACC	67.81(5)	89.29 (2)	73.59(7)	81.9 (6)	82.77 (6)	80.03 (5)	(5.17)
HLFR	43.74(8)	59.79 (8)	53.79(8)	46.02 (8)	55.67 (8)	66.57 (7)	(7.83)
ACDWM	72.23(1)	75.74 (7)	85.86(4)	94.86 (1)	83.28 (5)	81.64 (4)	(3.67)
SPECIAL	68.84(3)	89.81 (1)	91.72(3)	83.64 (4)	86.31 (3)	93.1 (2)	(2.67)

Table 4. F1-measure percentage and ranking of all algorithms on all datasets

Dataset	Weather	Electricity	SEA	Agrawal	RotHyperplane	RotChBoard-CDR	Avg. Rank
HT	52.36 (6)	77.15 (5)	91.04(2)	79.04 (5)	84.15 (4)	62.13 (8)	(5)
DWM	59.98 (2)	73.56 (6)	76.69(6)	82.43 (2)	89.86 (1)	71.87 (7)	(4)
OzaBagAdwin	50.45 (7)	79.86 (4)	91.85(1)	81.46 (3)	87.98 (2)	85.38 (3)	(3.33)
OzaBoostAdwin	57.59 (4)	85.92 (3)	81.22(5)	76.86 (7)	76.58 (7)	94.33 (1)	(4.5)
ADACC	57.29 (5)	87.73 (2)	68.33(7)	78.14 (6)	82.74 (6)	79.95 (6)	(5.33)
HLFR	28.45 (8)	54.25 (8)	42.67(8)	31.41 (8)	55.27 (8)	84.49 (4)	(7.33)
ACDWM	62.14 (1)	72.14 (7)	82.53(4)	91.77 (1)	83.28 (5)	80.51 (5)	(3.83)
SPECIAL	58.85 (3)	88.37 (1)	89.21(3)	80.21 (4)	86.28 (3)	93.1 (2)	(2.67)

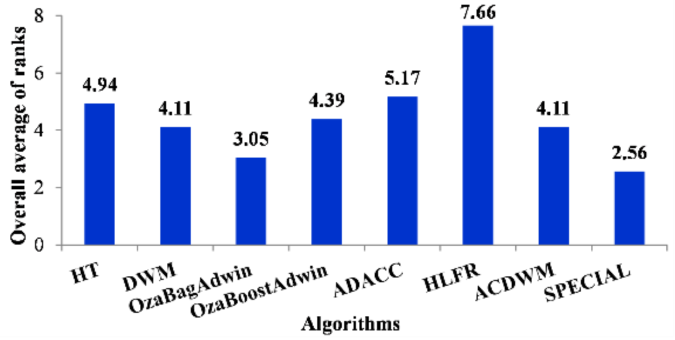

Figure 1. Overall average of ranks of all algorithms

3.2. Statistical results

The empirical results in Tables 2 to 4 show varying performances of different algorithms on different data sets. Hence, to rigorously assess the performances of all studied algorithms we carry out the nonparametric statistical tests as indorsed in [38]. We perform the Iman-Davenport test with a confidence of 95% (α=0.05) on all evaluation metrics of above mentioned eight algorithms. It rejects the null hypothesis (H0: Ranks of all algorithms are equivalent) for each evaluation metric and infers that at least one of the studied algorithms shows better performance than others on each measure. As the empirical result in Figure 1 claims that the proposed algorithm SPECIAL is the overall best performer, we conduct the pairwise Friedman posthoc test with Finner’s correction [38] to statistically analyse whether SPECIAL is the best performer among the other seven state-of-the-art classifiers on each metric. Table 5 summarizes the results of the posthoc test with a confidence of 95% (α=0.05). The bold-faced values indicate the noteworthy performance improvement of SPECIAL as compared to the other seven classifiers.

Table 5. Results of pairwise Friedman posthoc test (α=0.05) to compare SPECIAL on all metrics

Metric	HT	DWM	OzaBagAdwin	OzaBoostAdwin	ADACC	HLFR	ACDWM
Accuracy	0.2	0.2	0.9	0.2	0.2	**0**	0.2
G-mean	**0**	**0**	**0**	0.1	**0**	**0**	0.2
F1-measure	0.1	0.4	0.6	0.1	0.1	**0**	0.1

4. CONCLUSION

The proposed algorithm SPECIAL provides a novel joint solution to the challenging problem of learning in dynamic data streams with skewness and concept drifts. SPECIAL is a passive drift detection ensemble with the smartness of G-mean maximization and ageing-based adaptive learning. It integrates five seminal ensembles as its base learners. It forms the smart pools of data mapping to the same area of the feature space. These pools point to the local expertise sub ensembles which are likely to give the best classification results in that feature space. SPECIAL follows online learning with a test-then-train approach. It adapts to the dynamicity in data by employing an ageing-based strategy to forget the historic data and to emphasize the recent data. It handles skewness in data streams with the objective of G-mean maximization. The performance of SPECIAL is compared with seven state-of-the-art ensembles on three performance metrics–i) accuracy, ii) G-mean, and iii) F1-measure using a variety of benchmark datasets. Based on the empirical analysis of these metrics the overall average ranking of SPECIAL indicates that it outperforms the other state-of-the-art ensembles in adaptive learning of dynamic data streams. The statistical analysis underpins that the proposed online ensemble model shows noteworthy performance improvement. The current research presents a passive drift detection model of an ensemble of ensembles. The future study will explore the active drift detection model to handle different types of drifts. Also, the presented study assesses the performance of the proposed model empirically and statistically. So, in the future, we would like to focus on the theoretical performance guarantees of the SPECIAL algorithm.

REFERNCES

[1] N. Seman and N. Atiqah Razmi, "Machine learning-based technique for big data sentiments extraction," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 3, pp. 473–479, Sep. 2020, doi: 10.11591/ijai.v9.i3.pp473-479.

[2] J. Sun, H. Li, H. Fujita, B. Fu, and W. Ai, "Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting," *Information Fusion*, vol. 54, pp. 128–144, Feb. 2020, doi: 10.1016/j.inffus.2019.07.006.

[3] M. A. Rezvi, S. Moontaha, K. A. Trisha, S. T. Cynthia, and S. Ripon, "Data mining approach to analyzing intrusion detection of wireless sensor network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 1, pp. 516–523, Jan. 2021, doi: 10.11591/ijeecs.v21.i1.pp516-523.

[4] H. M. Gomes, J. P. Barddal, F. Enembreck, and A. Bifet, "A survey on ensemble learning for data stream classification," *ACM Computing Surveys*, vol. 50, no. 2, pp. 1–36, Jun. 2017, doi: 10.1145/3054925.

[5] L. Deshpande and M. N. Rao, "Concept drift identification using classifier ensemble approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 1, pp. 19–25, Feb. 2018, doi: 10.11591/ijece.v8i1.pp19-25.

[6] E. Alothali, H. Alashwal, and S. Harous, "Data stream mining techniques: a review," *TELKOMNIKA*, vol. 17, no. 2, pp. 728–737, 2019, doi: 10.12928/TELKOMNIKA.v17i2.11752.

[7] I. Khamassi, M. Sayed-Mouchaweh, M. Hammami, and K. Ghédira, "Discussion and review on evolving data streams and concept drift adapting," *Evolving Systems*, vol. 9, no. 1, pp. 1–23, Mar. 2018, doi: 10.1007/s12530-016-9168-2.

[8] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, May 2017, doi: 10.1016/j.eswa.2016.12.035.

[9] S. Ancy and D. Paulraj, "Handling imbalanced data with concept drift by applying dynamic sampling and ensemble classification model," *Computer Communications*, vol. 153, pp. 553–560, 2020, doi: 10.1016/j.comcom.2020.01.061.

[10] S. Ren *et al.*, "Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning," *Knowledge-Based Systems*, vol. 163, pp. 705–722, Jan. 2019, doi: 10.1016/j.knosys.2018.09.032.

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

[12] H. Ghomeshi, M. M. Gaber, and Y. Kovalchuk, "Ensemble dynamics in non-stationary data stream classification," in *Learning from Data Streams in Evolving Environments, Studies in Big Data*, vol. 41, M. Sayed-Mouchaweh, Ed. Springer, Cham, pp. 123–153, 2019.

[13] R. S. M. de Barros and S. G. T. de C. Santos, "An overview and comprehensive comparison of ensembles for concept drift," *Information Fusion*, vol. 52, pp. 213–244, Dec. 2019, doi: 10.1016/j.inffus.2019.03.006.

[14] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining-KDD '03*, 2003, Art. no. 226, doi: 10.1145/956750.956778.

[15] D. Brzezinski and J. Stefanowski, "Reacting to different types of concept drift: The accuracy updated ensemble algorithm," *IEEE*

Transactions on Neural Networks and Learning Systems, vol. 25, no. 1, pp. 81–94, Jan. 2014, doi: 10.1109/TNNLS.2013.2251352.

[16] Y. Lu, Y. Cheung, and Y. Y. Tang, "Dynamic weighted majority for incremental learning of imbalanced data streams with concept drift," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17)*, 2017, pp. 2393–2399, doi: 10.24963/ijcai.2017/333.

[17] Y. Lu, Y.-M. Cheung, and Y. Yan Tang, "Adaptive chunk-based dynamic weighted majority for imbalanced data streams with concept drift," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 2764–2778, Aug. 2020, doi: 10.1109/TNNLS.2019.2951814.

[18] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining-KDD '09*, 2009, Art. no. 139, doi: 10.1145/1557019.1557041.

[19] D. Brzezinski and J. Stefanowski, "Combining block-based and online methods in learning ensembles from concept drifting data streams," *Information Sciences*, vol. 265, pp. 50–67, May 2014, doi: 10.1016/j.ins.2013.12.011.

[20] J. Kolter and M. Maloof, "Dynamic Weighted Majority : An ensemble method for drifting concepts," *Journal of Machine Learning Research*, vol. 8, pp. 2755–2790, 2007.

[21] G. Jaber, A. Cornuéjols, and P. Tarroux, "A new on-line learning method for coping with recurring concepts: The ADACC system," in *Neural Information Processing (ICONIP 2013), Lecture Notes in Computer Science*, vol. 8227, M. Lee, A. Hirose, Z. Hou, and R. M. Kil, Eds. Daegu, Korea (Republic of): Springer, Berlin, Heidelberg, pp. 595–604. 2013.

[22] P. R. L. Almeida, L. S. Oliveira, A. S. Britto, and R. Sabourin, "Adapting dynamic classifier selection for concept drift," *Expert Systems with Applications*, vol. 104, pp. 67–85, Aug. 2018, doi: 10.1016/j.eswa.2018.03.021.

[23] A. Bifet, G. Holmes, B. Pfahringer, and R. Gavaldà, "Improving adaptive bagging methods for evolving data streams," in *Advances in Machine Learning, ACML 2009, Lecture Notes in Computer Science*, vol. 5828, Z. Zhou and T. Washio, Eds. Springer, Berlin, Heidelberg, pp. 23–37, 2009.

[24] R. S. M. Barros and S. G. T. C. Santos, "A large-scale comparison of concept drift detectors," *Information Sciences*, vol. 451–452, pp. 348–370, Jul. 2018, doi: 10.1016/j.ins.2018.04.014.

[25] D. R. de L. Cabral and R. S. M. de Barros, "Concept drift detection based on Fisher's Exact test," *Information Sciences*, vol. 442–443, pp. 220–234, May 2018, doi: 10.1016/j.ins.2018.02.054.

[26] S. Yu, Z. Abraham, H. Wang, M. Shah, Y. Wei, and J. C. Príncipe, "Concept drift detection and adaptation with hierarchical hypothesis testing," *Journal of the Franklin Institute*, vol. 356, no. 5, pp. 3187–3215, Mar. 2019, doi: 10.1016/j.jfranklin.2019.01.043.

[27] H. Ali, M. N. Mohd Salleh, R. Saedudin, K. Hussain, and M. F. Mushtaq, "Imbalance class problems in data mining: a review," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 3, p. 1552, Jun. 2019, doi: 10.11591/ijeecs.v14.i3.pp1552-1563.

[28] W. Zhang and J. Wang, "A hybrid learning framework for imbalanced stream classification," in *2017 IEEE International Congress on Big Data (BigData Congress)*, Jun. 2017, pp. 480–487, doi: 10.1109/BigDataCongress.2017.70.

[29] Y. Sun, "A novel ensemble classification for data streams with class imbalance and concept drift," *International Journal of Performability Engineering*, vol. 13, no. 6, pp. 945–955, 2017, doi: 10.23940/ijpe.17.06.p15.945955.

[30] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, Jan. 2018, doi: 10.1016/j.neucom.2017.06.084.

[31] N. C. Oza, "Online ensemble learning," University of California, Berkeley, 2001.

[32] R. Kulkarni, S. Revathy, and S. Patil, "A novel approach to maximize G-mean in nonstationary data with recurrent imbalance shifts," *The International Arab Journal of Information Technology*, vol. 18, no. 1, pp. 103–113, Dec. 2020, doi: 10.34028/iajit/18/1/12.

[33] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 1–1, Dec. 2018, doi: 10.1109/TKDE.2018.2876857.

[34] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive online analysis," *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010.

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

[36] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '00*, 2000, pp. 71–80, doi: 10.1145/347090.347107.

[37] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 2007 SIAM International Conference on Data Mining*, Apr. 2007, pp. 443–448, doi: 10.1137/1.9781611972771.42.

[38] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, May 2010, doi: 10.1016/j.ins.2009.12.010.

BIOGRAPHIES OF AUTHORS

Radhika Vikas Kulkarni received the M.Tech. degree in Computer Science and Technology from the Shivaji University, Kolhapur, India in 2011. She is currently pursuing a Ph.D. degree with the Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai India. She is working as an Assistant Professor in the Department of Information Technology, Pune Institute of Computer Technology, Pune, India. Her current research interests include Machine Learning, Data Analytics, and Big Data. Email: radhikavikaskulkarni@gmail.com.

S. Revathy is presently working as an Associate Professor in the Department of Information Technology, Sathyabama Institute of Science and Technology, Chennai India. Her research interest includes Machine Learning, Data Analytics, and Big Data. She has published over 20 papers in refereed journals. Email: revathy.it@sathyabama.ac.in.

Suhas Haribhau Patil received a Ph.D. degree in Computer Science and Engineering from Bharati Vidyapeeth Deemed University, Pune, India in 2009. He is currently working as a Professor in the Department of Computer Science and Engineering, Bharati Vidyapeeth Deemed University College of Engineering, Pune, India. His research area includes Machine Learning, Expert Systems, Computer Networks, Operating Systems, System Software. He has published over 65 papers in international journals, 36 papers in international conferences, and 42 papers in national conferences. Email: shpatil@bvucoep.edu.in.