International Journal on
Advanced Science
Engineering
Information Technology

Vol.11 (2021) No. 5
ISSN: 2088-5334

An Empirical Study of Online Learning in Non-stationary Data Streams Using Ensemble of Ensembles

Radhika V. Kulkarni [a,*], S. Revathy [a], Suhas H. Patil [b]

[a] *Department of Computer Science Engineering, Sathyabama Institute of Science and Technology, Chennai, 600119, India*
[b] *Department of Computer Science Engineering, Bharati Vidyapeeth's College of Engineering, Pune, 411043, India*
*Corresponding author: *radhikavikaskulkarni@gmail.com*

***Abstract*—Numerous information system applications produce a huge amount of non-stationary streaming data that demand real-time analytics. Classification of data streams engages supervised models to learn from a continuous infinite flow of labeled observations. The critical issue of such learning models is to handle dynamicity in data streams where the data instances undergo distributional change called concept drift. The online learning approach is essential to cater to learning in the streaming environment as the learning model is built and functional without the complete data for training in the beginning. Also, the ensemble learning method has proven to be successful in responding to evolving data streams. A multiple learner scheme boosts a single learner's prediction by integrating multiple base learners that outperform each independent learner. The proposed algorithm EoE (Ensemble of Ensembles) is an integration of ten seminal ensembles. It employs online learning with the majority voting to deal with the binary classification of non-stationary data streams. Utilizing the learning capabilities of individual sub ensembles and overcoming their limitations as an individual learner, the EoE makes a better prediction than that of its sub ensembles. The current communication empirically and statistically analyses the performance of the EoE on different figures of merits like accuracy, sensitivity, specificity, G-mean, precision, F1-measure, balanced accuracy, and overall performance measure when tested on a variety of real and synthetic datasets. The experimental results claim that the EoE algorithm outperforms its state-of-the-art independent sub ensembles in classifying non-stationary data streams.**

***Keywords*—Concept drift; data stream; ensemble; non-stationary data classification; online learning.**

Manuscript received 24 Sep. 2020; revised 28 Apr. 2021; accepted 7 May 2021. Date of publication 31 Oct. 2021.

I. INTRODUCTION

Recent developments in computational intelligence have focused on solving challenging environmental dynamicity-related issues. A wide variety of real-world applications like network intrusion detection [1], analysis of social media [2], [3], analysis of time-series data [4], condition-based maintenance [5], client credit analysis [6], financial risk prediction [7] need to process dynamic data received as streams. Non-stationary data streams characterize significant volumes of rapidly flowing boundless data [8], [9]. Such dynamic data streams may have uneven data samples referred to as class imbalance [10], [11] and changing data distribution referred to as concept drift [12], [13]. Thus, learning in a non-stationary environment is becoming an interesting and challenging research topic, and rich literature is available for the same [9], [12], [14]–[17].The streaming environment produces a quantum of data in sequence. Hence, the whole training data is not present in the beginning. The evolving nature of dynamic data streams demands online learning that builds and updates the learning model incrementally [18]. Ensemble learning is also an appropriate approach to learn in a non-stationary environment as the combination of different single learning models in an ensemble builds a superior learner by compensating for the weaknesses of its single learners [19].

Better generalization capability made ensembles more popular than a single classifier. A variety of ensemble learning algorithms for data stream mining have been published [17], [20]. Most of them like Weighted Majority Algorithm (WM) [21], Accuracy Weighted Ensemble (AWE) [22], Dynamic Weighted Majority (DWM) [23], Dynamic Classifier Selection (DCS) [24], Gradual Resampling Ensemble (GRE) [25], an ensemble based on Dynamic Classifier Selection (DCS) [24], Sample-based Online Learning Ensemble (SOLE) [6], and an ensemble for Handling Imbalanced Data with Concept Drift (HIDC) [10]. The HIDC handles the dynamicity in the data streams by including a new learner (and possibly removing an old one or the worst performing one) to develop an ensemble incrementally with each incoming data sample. These ensemble algorithms follow the passive detection of concept drifts. There is a variety of active drift detecting ensembles like BagADWIN [26], Adaptive Random Forest (ARF) [27], Wilcoxon Rank Sum Test Drift Detector (WSTD) [28], Dynamic AdaBoost.NC with Multiple Sub classifiers for Imbalance and Drifts (DAMSID) [5], Kappa Updated Ensemble (KUE) [29] that update the learning model on detection of a change in the data distribution. The SAM-kNN model [30] combines the concept of self-adjusting memory with the k-nearest-neighbor algorithm that creates ensemble members for the current and old knowledge base. A comprehensive study on ensemble learning for drifted skewed data is available in Priya and Uthra [31]. Various drift detection methods in data stream learning are categorized by Barros and Santos [19] and Hu *et al.* [32].

In a typical batch processing algorithm, all data instances for a learner's training are present at the start. In contrast, the incremental learner does not wait for all data instances for training to be received, and hence it is suitable for the streaming environment [18]. Online learning is incremental learning that uses each newly arrived example to test the learning and train the next model [18]. Oza [33] has introduced online bagging and boosting algorithms to learn in streaming data. Learn++.NSE [34], Online Accuracy Updated Ensemble (OAUE) [35], Reactive Drift Detection Method (RDDM) [36], Knowledge-Maximized Ensemble (KME) [37], Diversified Dynamic Weighted Majority (DDWM) [38],concept drift detection based on Online Sequential Extreme Learning Machine(OS-ELM) [39], Adaptive Chunk-based Dynamic Weighted Majority (ACDWM) [40], Dynamic Updated Ensemble (DUE) [41] are some of the examples of incremental learners.

This research aims to build an online ensemble learning model with the test and then train method to classify non-stationary data streams. This research employed ten seminal ensemble learning approaches for streaming data classification. Using these ensembles, we developed a majority voting-based ensemble of ensembles that follows online learning as described in Oza[33]. This empirical study follows the test-then-train approach to compare the proposed algorithm EoE (Ensemble of Ensembles) with state-of-the-art ensemble learning algorithms on different performance metrics. Also, the empirical analysis in this research is supported by statistical tests.

II. Materials and Method

The current section presents the problem description, the proposed methodology, and the experimental framework of the reported empirical study.

A. Problem Description

The problem being addressed in this research work is supervised binary classification on non-stationary streaming data. Formally, a data stream provides boundless data instances $\{d_1, d_2, d_3...\}$ where $d_t \in D = \mathbb{R}^k$ is a data instance in k-dimensional feature space received at time step $t = 1, 2, \ldots$. An input data instance d_t is initially unlabelled and its class label $c_t \in C = \{c_1, c_2, \ldots, c_L\}$ where L is the number of classes received at a constant amount of time $t' \in \mathbb{R}^+$ after d_t. A framework of our problem considers $c_t \in C = \{0, 1\}$ where the class label '*0*' represents a negative class, and a class label '*1*' represents a positive class. Also, it considers class imbalance where the instances of a negative class, say D^0 are in the majority and the instances of a positive class, say D^1 are in the minority i.e., $|D^0| >> |D^1|$. Let a joint distribution $P_t(D, C)$ define a concept that produces a tuple (d_t, c_t) at step t. This framework allows non-stationary data where the concept may change at time t i.e., $P_{t-1}(D, C) \neq P_t(D, C)$ [12].

The current work aims to design an online classifier $f : D \rightarrow C$ to predict a class label c_t is associated with an input data instance d_t. In the streaming environment, data is received sequentially so an online learning model gets evolved with the latest incoming data instances [12], [18].

B. Proposed Methodology

To cater to the classification of non-stationary data streams, this research contributes to the following ways:

- Bagging using ten seminal ensembles as base learners with the test-then-train approach for dynamic data stream classification.
- Online ensemble learning builds the model on every incoming sample without waiting for the whole set of training samples.
- Empirical and statistical analysis for comparing the performance of the proposed ensemble with other state-of-the-art ensembles.

As ensemble integrates reasonably well-performing but a variety of base learners, we build the proposed bagging model EoE (Ensemble of Ensembles) using the following ten seminal ensembles as base learners:

ADWIN Bagging (BagADWIN) [26]extends the online bagging algorithm [33] by incorporating the ADWIN (ADaptive sliding WINdowing) algorithm [42]. In BagADWIN, the ADWIN algorithm detects the change and estimates the learner's weightage. Accordingly, the worst performer in an ensemble of learners is deleted, and a new learner is included in an ensemble.

ADWIN Boosting (BoostADWIN) [26] combines the online boosting algorithm [33] with ADWIN (ADaptive sliding WINdowing) algorithm [42] for the detection of a concept change.

Weighted Majority Algorithm (WM) [21] is an ensemble of weighted experts. Depending on the weights of the base learners that have accurate classification results and the weights of the base learners that misclassify, it computes the weighted majority and accordingly produces the final classification result of the ensemble.

Dynamic Weighted Majority (DWM) [23] is a revised version of WM [21]. Irrespective of the prediction result of an ensemble, the weight of an individual base learner is reduced by a fixed value if it misclassifies. Considering the global prediction result an ensemble is updated continuously. It removes a base learner with weight below a threshold.

Anticipative Dynamic Adaptation to Concept Change (ADACC) [43] maintains a pool of incremental learning models. It evaluates the prediction performance of each of the base learners in a pool for every τ time step. Out of the worst half base learners of a pool, it randomly selects the one worst performer and substitutes it with a new base learner. It protects the newly inserted base learner from removal for a certain time slot. The latest best-performing base learner in a pool provides the final prediction.

Hoeffding Tree (HT) [44] provides an incrementally developed tree that processes each instance with constant time and memory. It employs Hoeffding bound to define the count of necessary samples at each node of a tree.

Leveraging Bagging (LevBag) [45] modifies BagADWIN [26] by setting the higher value of diversity factor (λ=6) in Poisson approximation for detecting the probability of training samples. It also applies ensemble output randomization. Using majority voting LevBag determines the final classification result of an ensemble.

Accuracy Weighted Ensemble (AWE) [22]trains a new learner on each incoming batch of data instances and tests all existing learners on that batch. The weightage to the base learner's vote is based on its prediction accuracy. The most accurate k learners form a new ensemble.

Accuracy Updated Ensemble (AUE) [46] builds an ensemble using chunk-based incremental weighted learners. Applying weighted voting on classification results of all base learners on the recent data chunk AUE computes the final prediction of an ensemble. A new learner built on each new block of data replaces the worst performer in the ensemble.

Online Accuracy Updated Ensemble (OAUE) [35] follows an online learning approach in which it evaluates every base learner and computes its accuracy-based weight after every observation. It periodically deletes the least accurate learner from the ensemble.

Bagging works better with unstable learners [33], so we implemented EoE bagging using Hoeffding decision trees [44] as base learners in all its sub ensembles. Thus, the proposed EoE gives better performance as verified under Section III. We apply maximum voting for the final prediction.

We follow the online learning approach by Oza [33] that simulates the idea of bootstrapping for streaming data. Generally, bagging demands the whole set of n training samples to have bootstrapping with replacement. Accordingly, bootstrapped samples follow a binomial distribution where X is the number of copies of each data sample. It is given by equation 1. However, the whole set of training samples is initially unavailable in the streaming environment, and the flow of incoming samples continues boundlessly. The online learning strategy in [33] approximates the binomial distribution of replicated training instances by Poisson(1) distribution when the training data size $n \to \infty$. It is defined by equation 2.

$$P(X=k) = \binom{n}{k} (1/n)^k \big(1-(1/n)\big)^{n-k} \qquad (1)$$

$$P(X=k) = e^{-1}/(k!) \qquad (2)$$

In streaming data, as we cannot get all data samples initially, we initially do not have data size n. Hence, we prefer the evaluation of EoE through the test-then-train approach instead of the holdout or cross-validation method. Accordingly, the EoE model first employs testing on a newly arrived sample before using it for the model's training. Thus, the EoE maximally utilizes the available data samples and evaluates the model incrementally through the test-then-train approach. The proposed algorithm Ensemble of Ensembles (EoE) is described below.

```
Algorithm EoE:
Input: $(d_t, c_t)$ is an incoming data instance at time $t = \{1, 2, \ldots\}$ of data
        stream $D$, $S = \{s \in \{1,2,.., E\}\}$ is an ensemble of ensembles, $E$ is
        the number of sub ensembles working as base learners in S.
Output: A composite hypothesis
        $H(d_t) = \arg\max_{c_t \in \{0,1\}} \sum_{s=1}^{E} I(h_t(d_t) = c_t)$
Initialize: E=10
Do for each incoming instance $(d_t, c_t)$
    for $s$ = 1 to $E$ do
        Let $r \sim Poisson(1)$
        By the test-then-train approach repeat $r$ times
            training of the base learner $s$; $(h_t(d_t) = c_t)$;
    end for
End
```

Fig. 1 presents the flowchart of the EoE algorithm. On the arrival of each new data sample, the EoE model always gets being tested on that unseen sample. The evaluation phase compares the prediction result with the actual class label of the data sample. It empirically and statistically analyses the performance of the EoE on different metrics. After the testing phase, the same data sample is used to train the EoE model. It follows bootstrapping by Poisson approximation. This trained model is then used to test the next incoming new sample.

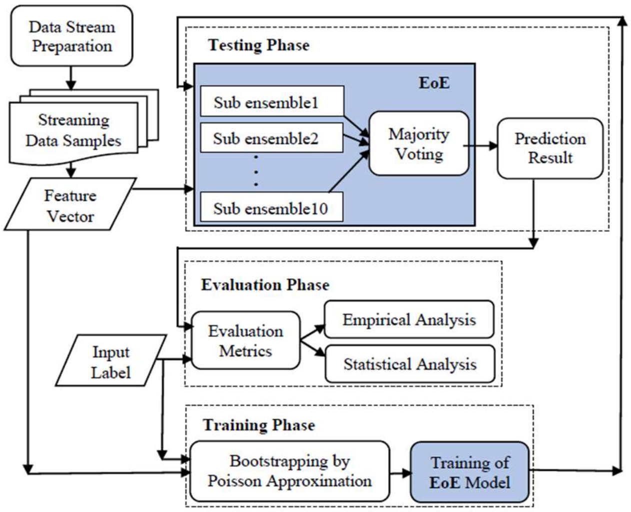

Fig. 1 The flowchart of the EoEalgorithm

TABLE I
SUMMARY OF DATASETS USED

Dataset	# Instances	# Features	# Classes	Positive %	Negative %	Type
SEA	50000	3	2	37.16	62.84	Synthetic
Agrawal	100000	9	2	32.66	67.34	Synthetic
Rotating Hyperplane	200000	10	2	49.97	50.03	Synthetic
Rotating Checkerboard-CDR	409600	2	2	49.99	50.01	Synthetic
Weather	18159	8	2	31.38	68.62	Real
Electricity	45312	8	2	42.45	57.55	Real
Airlines	539383	7	2	44.54	55.46	Real
KDD Cup 10 percent	494000	41	2	23.14	76.86	Real

C. Dataset Description

There are publicly available numerous real and synthetic datasets widely used for concept drift and imbalance learning, as listed in Lu *et al.*[13]. The experimentation is carried out on eight popular benchmark datasets, out of which four are synthetic, and four are real datasets. The selected data sets differ in sample size, class distribution, and imbalance ratio to ensure a detailed evaluation of results. Many of these datasets like Agrawal, Rotating Hyperplane, Electricity pricing, Airlines are available with MOA (Massive Online Analysis) framework [47]. SEA, Weather, KDD Cup 10 percent, and Rotating Checker Board with Constant Drift Rate (Rotating Checker Board-CDR) datasets are taken from the repository [34]. Table I presents the summary of these datasets employed for the experimentation.

D. Evaluation Metrics

The confusion matrix showed in Table II illustrates the result of binary classification. Let ‘0’ define a negative class and ‘1’ define a positive class.

TABLE II
CONFUSION MATRIX FOR BINARY CLASSIFICATION

	Actual ‘0’	**Actual ‘1’**
Predicted ‘0’	TN (True Negative)	FN (False Negative)
Predicted ‘1’	FP (False Positive)	TP (True Positive)

Non-stationary data streams may possess concept drifts and skewness in it. If data possess skewness, then accuracy measurement will not evaluate a learner’s performance correctly. It may reflect the better accuracy due to correct classification of the majority samples hiding the poor performance of the learner in classifying the minority samples [11], [12]. Therefore, the current empirical study employs different figures of merits to thoroughly evaluate the performance of the algorithms mentioned above in classifying non-stationary data streams. There are other different popular evaluation metrics like sensitivity, specificity, G-mean, precision, F1-measure, and balanced accuracy. A learner’s classification performance for both majority and minority samples are assessed through metrics such as sensitivity, specificity, G-mean, and balanced accuracy.

Additionally, this study presents a combination of evaluation metrics as “Overall Performance Measure” (OPM), which computes a single value to reflect how well an algorithm performs on a specific dataset. It is a convex function of accuracy, G-mean, and F1-measure. In OPM computation, we have considered G-mean and not sensitivity, specificity, and balanced accuracy as G-mean considers sensitivity and specificity. Also, instead of precision, we have considered only F1-measure in OPM calculation as it also reflects precision. The following equations 3 to 10 define different evaluation metrics.

$$Accuracy = ((TP + TN)/(TP + TN + FP + FN)) \quad (3)$$

$$Sensitivity = (TP/(TP + FN)) \quad (4)$$

$$Specificity = (TN/(TN + FP)) \quad (5)$$

$$G - mean = \left(\sqrt{Sensitivity \cdot Specificity}\right) \quad (6)$$

$$Pr\ e cision = (TP/(TP + FP)) \quad (7)$$

$$F1 - measure = \left(\frac{2 \cdot Sensitivity \cdot Pr\ e cision}{Sensitivity + Pr\ e cision}\right) \quad (8)$$

$$BalancedAccuracy = ((Sensitivity + Specificity)/2) \quad (9)$$

$$OPM = \frac{1}{\lambda_1} \cdot Accuracy + \frac{1}{\lambda_2} \cdot G - mean + \frac{1}{\lambda_3} \cdot F1 - measure$$

Where

$$\frac{1}{\lambda_1} = \frac{1}{\lambda_2} = \frac{1}{\lambda_3} = \frac{1}{3} \quad (10)$$

E. Experimental Setup

The proposed algorithm EoE is compared with ten state-of-the-art algorithms. We have used the Java-based MOA framework for the empirical study and refer to state-of-the-art ensembles implemented in it with default settings. All data instances are initially unavailable for training in a streaming environment, so the current online learning experiments follow the test-then-train approach. In online learning, the order of incoming data instances affects a learner’s performance[33], so we test all the algorithms independently on ten different orderings of each dataset. We average the results of such ten experiments per dataset using all eleven algorithms. The study compares the performances of algorithms by ranking these averages from (1) to (11). The lower the value of the rank, the better is the performance of the algorithm. Thus, a rank (1) indicates the best, and a rank (11) indicates the worst performer among eleven algorithms.

III. RESULTS AND DISCUSSION

This section investigates the empirical and statistical tests to evaluate the performance of EoE.

A. Empirical Results

The empirical study of online learning of non-stationary data streams using an ensemble of ensembles employs eight figures of merits: - 1) Accuracy (Acc), 2) Sensitivity (Se), 3) Specificity (Sp), 4) G-mean, 5) Precision (Prec), 6) F1-Measure (F1), 7) Balanced Accuracy (Bal Acc), and 8) Overall Performance Measure (OPM). All empirical results of performance measurement are presented in percentages. Tables III-VI present the average results of the figures of merits on four synthetic datasets, and tables VII-X present those results on four real datasets.

The experimental results on the SEA dataset are recorded in Table III. LevBag has the best sensitivity, G-mean, F1-measure, balanced accuracy, and OPM when tested on the SEA dataset. However, EoE is the best performer because of the overall average ranking. Table IV presents the experimental results on the Agrawal dataset. The results show that EoE is the best performer based on accuracy, G-mean, F1-measure, balanced accuracy, OPM, and overall average ranking, while ADACC is the worst performer on the Agrawal dataset.

TABLE III
SEA Dataset Empirical Results and Ranking Summary

	Bag ADWIN	Boost ADWIN	DWM	ADACC	HT	LevBag	AWE	WM	AUE	OAUE	EoE
Acc	85.46 ± 0.54 (4)	81.08 ± 0.92 (11)	85.43 ± 0.86 (5)	82.35 ± 0.86 (10)	85.2 ± 0.07 (8)	85.74 ± 0.59 (2)	85.15 ± 0.56 (9)	85.34 ± 0.09 (6)	85.34 ± 0.49 (7)	85.49 ± 0.72 (3)	85.74 ± 0.63 (1)
Se	75.52 ± 0.53 (4)	71.63 ± 0.93 (10)	73.1 ± 1.24 (8)	70.05 ± 1.16 (11)	75.32 ± 0.58 (5)	77.44 ± 0.58 (1)	72.5 ± 0.58 (9)	73.95 ± 0.45 (7)	75.02 ± 0.31 (6)	75.61 ± 0.62 (3)	75.81 ± 0.49 (2)
Sp	91.34 ± 0.63 (6)	86.67 ± 0.94 (11)	92.72 ± 0.64 (1)	89.63 ± 0.71 (10)	91.05 ± 0.34 (8)	90.65 ± 0.64 (9)	92.63 ± 0.58 (2)	92.09 ± 0.28 (3)	91.45 ± 0.61 (5)	91.33 ± 0.79 (7)	91.62 ± 0.72 (4)
G-mean	83.06 ± 0.53 (4)	78.79 ± 0.92 (11)	82.33 ± 0.97 (8)	79.24 ± 0.95 (10)	82.81 ± 0.18 (6)	83.78 ± 0.58 (1)	81.95 ± 0.56 (9)	82.52 ± 0.15 (7)	82.83 ± 0.43 (5)	83.1 ± 0.69 (3)	83.34 ± 0.59 (2)
Prec	83.77 ± 1.06 (6)	76.07 ± 1.52 (11)	85.58 ± 1.29 (1)	79.98 ± 1.34 (10)	83.27 ± 0.43 (8)	83.05 ± 1.06 (9)	85.34 ± 1.08 (2)	84.68 ± 0.4 (3)	83.85 ± 1.03 (5)	83.77 ± 1.38 (7)	84.26 ± 1.25 (4)
F1	79.43 ± 0.72 (4)	73.78 ± 1.19 (11)	78.85 ± 1.26 (8)	74.69 ± 1.22 (10)	79.09 ± 0.16 (6)	80.14 ± 0.78 (1)	78.4 ± 0.77 (9)	78.95 ± 0.15 (7)	79.19 ± 0.62 (5)	79.48 ± 0.95 (3)	79.81 ± 0.83 (2)
Bal Acc	83.43 ± 0.53 (4)	79.15 ± 0.91 (11)	82.91 ± 0.93 (8)	79.84 ± 0.91 (10)	83.18 ± 0.14 (6)	84.04 ± 0.58 (1)	82.56 ± 0.56 (9)	83.02 ± 0.12 (7)	83.24 ± 0.45 (5)	83.47 ± 0.69 (3)	83.71 ± 0.6 (2)
OPM	82.65 ± 0.6 (4)	77.89 ± 1.01 (11)	82.2 ± 1.03 (8)	78.76 ± 1.01 (10)	82.37 ± 0.14 (6)	83.22 ± 0.65 (1)	81.83 ± 0.63 (9)	82.27 ± 0.13 (7)	82.45 ± 0.51 (5)	82.69 ± 0.78 (3)	82.96 ± 0.68 (2)
AvgRank	(4.5)	(10.88)	(5.88)	(10.13)	(6.63)	(3.13)	(7.25)	(5.88)	(5.38)	(4)	(2.38)

TABLE IV
Agrawal Dataset Empirical Results and Ranking Summary

	BagADWIN	Boost ADWIN	DWM	ADACC	HT	LevBag	AWE	WM	AUE	OAUE	EoE
Acc	94.74 ± 0.08 (3)	92.76 ± 1.63 (9)	88.02 ± 0.19 (10)	85.46 ± 0.97 (11)	94.45 ± 0.21 (7)	94.76 ± 0.26 (2)	93.82 ± 0.84 (8)	94.49 ± 0.19 (6)	94.65 ± 0.11 (5)	94.68 ± 0.08 (4)	94.9 ± 0.05 (1)
Se	92.74 ± 0.38 (3)	88.29 ± 3.57 (9)	63.84 ± 0.46 (10)	62.91 ± 2.36 (11)	92.68 ± 0.92 (4)	92.4 ± 0.65 (8)	93.02 ± 1.94 (1)	92.48 ± 0.95 (6)	92.53 ± 0.34 (5)	92.45 ± 0.57 (7)	92.75 ± 0.29 (2)
Sp	95.71 ± 0.25 (6)	94.92 ± 0.75 (10)	99.74 ± 0.08 (1)	96.39 ± 0.39 (2)	95.3 ± 0.62 (9)	95.9 ± 0.67 (4)	94.22 ± 0.65 (11)	95.46 ± 0.61 (8)	95.67 ± 0.32 (7)	95.76 ± 0.38 (5)	95.94 ± 0.2 (3)
G-mean	94.21 ± 0.09 (2)	91.53 ± 2.21 (9)	79.8 ± 0.31 (10)	77.86 ± 1.6 (11)	93.98 ± 0.24 (6)	94.13 ± 0.1 (3)	93.61 ± 1.1 (8)	93.96 ± 0.25 (7)	94.09 ± 0.05 (5)	94.09 ± 0.11 (4)	94.33 ± 0.06 (1)
Prec	91.29 ± 0.44 (5)	89.36 ± 1.83 (10)	99.16 ± 0.25 (1)	89.41 ± 1.34 (9)	90.56 ± 1.06 (8)	91.63 ± 1.15 (3)	88.64 ± 1.19 (11)	90.83 ± 1.04 (7)	91.21 ± 0.56 (6)	91.38 ± 0.65 (4)	91.73 ± 0.36 (2)
F1	92.01 ± 0.1 (2)	88.81 ± 2.7 (9)	77.68 ± 0.4 (10)	73.84 ± 2.04 (11)	91.6 ± 0.27 (7)	92.01 ± 0.31 (3)	90.77 ± 1.31 (8)	91.64 ± 0.25 (6)	91.86 ± 0.13 (5)	91.91 ± 0.07 (4)	92.23 ± 0.06 (1)
Bal Acc	94.23 ± 0.09 (2)	91.61 ± 2.12 (9)	81.79 ± 0.26 (10)	79.65 ± 1.32 (11)	93.99 ± 0.23 (6)	94.15 ± 0.1 (3)	93.62 ± 1.08 (8)	93.97 ± 0.24 (7)	94.1 ± 0.04 (5)	94.11 ± 0.1 (4)	94.35 ± 0.05 (1)
OPM	93.65 ± 0.09 (2)	91.03 ± 2.18 (9)	81.83 ± 0.3 (10)	79.05 ± 1.53 (11)	93.34 ± 0.24 (7)	93.63 ± 0.22 (3)	92.73 ± 1.08 (8)	93.36 ± 0.23 (6)	93.53 ± 0.1 (5)	93.56 ± 0.09 (4)	93.82 ± 0.06 (1)
AvgRank	(3.13)	(9.25)	(7.75)	(9.63)	(6.75)	(3.63)	(7.88)	(6.63)	(5.38)	(4.5)	(1.5)

TABLE V
Rotating Hyperplane Dataset Empirical Results and Ranking Summary

	BagADWIN	Boost ADWIN	DWM	ADACC	HT	LevBag	AWE	WM	AUE	OAUE	EoE
Acc	81.07 ± 2.44 (5)	71.33 ± 2.01 (11)	81.5 ± 2.94 (3)	73.29 ± 3.33 (10)	80.4 ± 1.3 (9)	80.72 ± 2.16 (8)	81.64 ± 2.53 (1)	81.35 ± 1.16 (4)	80.89 ± 2.45 (7)	80.93 ± 2.52 (6)	81.55 ± 2.5 (2)
Se	80.96 ± 2.42 (5)	70.66 ± 2.25 (11)	81.5 ± 2.95 (3)	73.24 ± 3.3 (10)	80.33 ± 1.5 (9)	80.58 ± 2.13 (8)	81.5 ± 2.49 (2)	81.35 ± 1.33 (4)	80.64 ± 2.49 (7)	80.68 ± 2.55 (6)	82.31 ± 2.68 (1)
Sp	81.19 ± 2.46 (4)	71.99 ± 1.92 (11)	81.5 ± 2.93 (2)	73.33 ± 3.37 (10)	80.47 ± 1.1 (9)	80.86 ± 2.19 (7)	81.78 ± 2.57 (1)	81.34 ± 1 (3)	81.13 ± 2.42 (6)	81.19 ± 2.5 (5)	80.78 ± 2.33 (8)
G-mean	81.07 ± 2.44 (5)	71.32 ± 2.02 (11)	81.5 ± 2.94 (3)	73.29 ± 3.33 (10)	80.4 ± 1.3 (9)	80.72 ± 2.16 (8)	81.64 ± 2.53 (1)	81.35 ± 1.16 (4)	80.88 ± 2.45 (7)	80.93 ± 2.52 (6)	81.54 ± 2.5 (2)
Prec	81.13 ± 2.46 (4)	71.58 ± 1.96 (11)	81.48 ± 2.93 (2)	73.28 ± 3.36 (10)	80.41 ± 1.17 (9)	80.78 ± 2.19 (8)	81.71 ± 2.57 (1)	81.32 ± 1.05 (3)	81.01 ± 2.44 (7)	81.07 ± 2.52 (5)	81.04 ± 2.34 (6)
F1	81.13 ± 2.46 (5)	71.58 ± 1.96 (11)	81.48 ± 2.93 (3)	73.28 ± 3.36 (10)	80.41 ± 1.17 (9)	80.78 ± 2.19 (8)	81.71 ± 2.57 (2)	81.32 ± 1.05 (4)	81.01 ± 2.44 (7)	81.07 ± 2.52 (6)	81.04 ± 2.34 (1)
Bal Acc	81.04 ± 2.44 (5)	71.12 ± 2.07 (11)	81.49 ± 2.94 (3)	73.26 ± 3.33 (10)	80.37 ± 1.33 (9)	80.68 ± 2.16 (8)	81.6 ± 2.53 (1)	81.34 ± 1.19 (4)	80.83 ± 2.46 (7)	80.87 ± 2.53 (6)	81.67 ± 2.5 (2)
OPM	81.07 ± 2.44 (5)	71.33 ± 2.02 (11)	81.5 ± 2.94 (3)	73.29 ± 3.33 (10)	80.4 ± 1.3 (9)	80.72 ± 2.16 (8)	81.64 ± 2.53 (1)	81.35 ± 1.16 (4)	80.89 ± 2.45 (7)	80.93 ± 2.52 (6)	81.55 ± 2.5 (2)
AvgRank	(4.75)	(11)	(2.75)	(10)	(9)	(7.88)	(1.25)	(3.75)	(6.88)	(5.75)	(3)

TABLE VI
ROTATING CHECKERBOARD-CDR DATASET EMPIRICAL RESULTS AND RANKING SUMMARY

	BagAD WIN	Boost ADWIN	DWM	ADACC	HT	LevBag	AWE	WM	AUE	OAUE	EoE
Acc	85.46 ± 0.16 (6)	94.25 ± 0.19 (1)	71.95 ± 0.01 (8)	80.15 ± 0.11 (7)	60.38 ± 0.78 (10)	93.64 ± 0.03 (2)	52.65 ± 0.26 (11)	62.36 ± 1.11 (9)	86.58 ± 0.19 (5)	91.43 ± 0.1 (3)	90.33 ± 0.19 (4)
Se	85.83 ± 0.23 (6)	94.06 ± 0.26 (1)	71.67 ± 0.02 (8)	79.78 ± 0.29 (7)	59.77 ± 4.07 (10)	93.64 ± 0.05 (2)	53.27 ± 0.75 (11)	61.5 ± 2.71 (9)	86.42 ± 0.36 (5)	91.44 ± 0.13 (4)	92.27 ± 0.28 (3)
Sp	85.09 ± 0.26 (6)	94.43 ± 0.14 (1)	72.23 ± 0.02 (8)	80.52 ± 0.33 (7)	60.98 ± 4.65 (10)	93.64 ± 0.05 (2)	52.04 ± 0.65 (11)	63.22 ± 3.78 (9)	86.74 ± 0.26 (5)	91.42 ± 0.15 (3)	88.4 ± 0.37 (4)
G-mean	85.46 ± 0.16 (6)	94.25 ± 0.19 (1)	71.95 ± 0.01 (8)	80.14 ± 0.11 (7)	60.24 ± 0.74 (10)	93.64 ± 0.03 (2)	52.65 ± 0.26 (11)	62.28 ± 1.08 (9)	86.58 ± 0.19 (5)	91.43 ± 0.1 (3)	90.31 ± 0.19 (4)
Prec	85.19 ± 0.22 (6)	94.41 ± 0.15 (1)	72.06 ± 0.01 (8)	80.37 ± 0.23 (7)	60.59 ± 1.46 (10)	93.63 ± 0.04 (2)	52.61 ± 0.25 (11)	62.64 ± 1.72 (9)	86.69 ± 0.22 (5)	91.42 ± 0.14 (3)	88.83 ± 0.31 (4)
F1	85.51 ± 0.15 (6)	94.23 ± 0.2 (1)	71.86 ± 0.01 (8)	80.07 ± 0.11 (7)	60.08 ± 1.59 (10)	93.64 ± 0.03 (2)	52.94 ± 0.44 (11)	62.01 ± 1.15 (9)	86.56 ± 0.2 (5)	91.43 ± 0.1 (3)	90.51 ± 0.18 (4)
Bal Acc	85.46 ± 0.16 (6)	94.25 ± 0.19 (1)	71.95 ± 0.01 (8)	80.15 ± 0.11 (7)	60.38 ± 0.77 (10)	93.64 ± 0.03 (2)	52.65 ± 0.26 (11)	62.36 ± 1.11 (9)	86.58 ± 0.19 (5)	91.43 ± 0.1 (3)	90.33 ± 0.19 (4)
OPM	85.48 ± 0.15 (6)	94.24 ± 0.19 (1)	71.92 ± 0.01 (8)	80.12 ± 0.11 (7)	60.23 ± 1.03 (10)	93.64 ± 0.03 (2)	52.75 ± 0.32 (11)	62.22 ± 1.11 (9)	86.57 ± 0.19 (5)	91.43 ± 0.1 (3)	90.38 ± 0.19 (4)
AvgRank	(6)	(1)	(8)	(7)	(10)	(2)	(11)	(9)	(5)	(3.13)	(3.88)

TABLE VII
WEATHER DATASET EMPIRICAL RESULTS AND RANKING SUMMARY

	BagAD WIN	BoostA DWIN	DWM	ADACC	HT	LevBag	AWE	WM	AUE	OAUE	EoE
Acc	73.84 ± 0.66 (5)	72.67 ± 0.75 (7)	67.42 ± 1.08 (11)	67.64 ± 2.1 (9)	72.66 ± 0.76 (8)	77.11 ± 0.58 (1)	67.57 ± 1 (10)	72.7 ± 0.73 (6)	75.1 ± 0.5 (4)	75.89 ± 0.62 (3)	76 ± 0.93 (2)
Se	37.16 ± 3.68 (11)	53.36 ± 2.12 (5)	63.37 ± 5.67 (2)	54.85 ± 0.71 (4)	38.8 ± 5.72 (9)	48.42 ± 2.57 (6)	66.11 ± 0.47 (1)	39.18 ± 5.36 (8)	38.33 ± 3.86 (10)	42.82 ± 2.76 (7)	56.05 ± 1.72 (3)
Sp	90.61 ± 1.36 (3)	81.5 ± 1.19 (8)	69.27 ± 2.83 (10)	73.48 ± 2.79 (9)	88.15 ± 2.89 (5)	90.23 ± 0.69 (4)	68.24 ± 1.55 (11)	88.02 ± 2.76 (6)	91.91 ± 1.68 (1)	91 ± 0.86 (2)	85.12 ± 1.44 (7)
G-mean	57.94 ± 2.48 (11)	65.92 ± 1.15 (5)	66.13 ± 2.1 (3)	63.48 ± 1.55 (6)	58.27 ± 3.53 (10)	66.07 ± 1.58 (4)	67.16 ± 0.67 (2)	58.54 ± 3.27 (9)	59.27 ± 2.36 (8)	62.39 ± 1.78 (7)	69.06 ± 1 (1)
Prec	64.5 ± 1.89 (4)	56.89 ± 1.36 (8)	48.54 ± 1.24 (11)	48.75 ± 3.31 (10)	60.34 ± 2.87 (6)	69.4 ± 1.09 (1)	48.79 ± 1.21 (9)	60.28 ± 2.76 (7)	68.64 ± 2.41 (2)	68.55 ± 1.46 (3)	63.34 ± 2.04 (5)
F1	47.03 ± 2.83 (10)	55.04 ± 1.39 (4)	54.87 ± 2.56 (5)	51.59 ± 2.03 (7)	46.88 ± 3.81 (11)	57 ± 1.83 (2)	56.13 ± 0.73 (3)	47.19 ± 3.52 (9)	49.03 ± 2.55 (8)	52.67 ± 2.08 (6)	59.44 ± 1.3 (1)
Bal Acc	63.88 ± 1.34 (9)	67.43 ± 0.92 (3)	66.32 ± 1.83 (6)	64.17 ± 1.7 (8)	63.47 ± 1.62 (11)	69.33 ± 1.07 (2)	67.17 ± 0.69 (4)	63.6 ± 1.51 (10)	65.12 ± 1.22 (7)	66.91 ± 1.12 (5)	70.59 ± 0.89 (1)
OPM	59.61 ± 1.99 (9)	64.54 ± 1.1 (3)	62.81 ± 1.91 (6)	60.9 ± 1.89 (8)	59.27 ± 2.7 (11)	66.73 ± 1.33 (2)	63.62 ± 0.8 (5)	59.47 ± 2.51 (10)	61.13 ± 1.8 (7)	63.65 ± 1.49 (4)	68.17 ± 1.08 (1)
AvgRank	(7.75)	(5.38)	(6.75)	(7.63)	(8.88)	(2.75)	(5.63)	(8.13)	(5.88)	(4.63)	(2.63)

TABLE VIII
ELECTRICITY DATASET EMPIRICAL RESULTS AND RANKING SUMMARY

	BagAD WIN	Boost ADWIN	DWM	ADACC	HT	LevBag	AWE	WM	AUE	OAUE	EoE
Acc	77.39 ± 2.21 (4)	77.18 ± 3.93 (5)	74.28 ± 1.68 (10)	73.26 ± 5.76 (11)	76.22 ± 1.49 (8)	80.35 ± 3.18 (1)	74.91 ± 1.05 (9)	76.33 ± 1.7 (7)	76.39 ± 0.32 (6)	78 ± 3.34 (3)	78.42 ± 2.78 (2)
Se	63.8 ± 4.51 (5)	72.66 ± 4.59 (1)	53.95 ± 5.34 (11)	60.59 ± 9.3 (8)	63.47 ± 5.61 (6)	70.69 ± 5.62 (2)	56.63 ± 1.28 (10)	63.15 ± 4.6 (7)	60.15 ± 3.32 (9)	63.88 ± 6.87 (4)	64.63 ± 6.1 (3)
Sp	87.41 ± 0.96 (7)	80.52 ± 3.51 (11)	89.27 ± 1.29 (1)	82.61 ± 3.16 (10)	85.63 ± 1.86 (9)	87.48 ± 1.44 (6)	88.4 ± 2.48 (4)	86.05 ± 1.2 (8)	88.37 ± 1.95 (5)	88.42 ± 0.82 (3)	88.6 ± 0.5 (2)
G-mean	74.65 ± 2.76 (5)	76.48 ± 4.04 (2)	69.31 ± 2.79 (11)	70.66 ± 6.55 (10)	73.63 ± 2.38 (7)	78.6 ± 3.65 (1)	70.73 ± 0.64 (9)	73.66 ± 2.38 (6)	72.86 ± 1.09 (8)	75.08 ± 4.15 (4)	75.61 ± 3.54 (3)
Prec	78.85 ± 1.88 (5)	73.36 ± 4.74 (10)	78.79 ± 1.08 (6)	71.68 ± 5.89 (11)	76.57 ± 1.15 (9)	80.55 ± 2.71 (2)	78.43 ± 2.96 (7)	76.95 ± 1.33 (8)	79.35 ± 1.59 (4)	80.13 ± 2.33 (3)	80.6 ± 1.7 (1)
F1	70.49 ± 3.32 (5)	73 ± 4.64 (2)	63.92 ± 3.43 (11)	65.61 ± 7.77 (10)	69.27 ± 2.94 (7)	75.26 ± 4.27 (1)	65.72 ± 0.76 (9)	69.3 ± 2.89 (6)	68.34 ± 1.36 (8)	71 ± 4.92 (4)	71.66 ± 4.19 (3)
Bal Acc	75.61 ± 2.5 (5)	76.59 ± 4.01 (3)	71.61 ± 2.15 (10)	71.6 ± 6.23 (11)	74.55 ± 2 (7)	79.08 ± 3.49 (1)	72.51 ± 0.83 (9)	74.6 ± 2.06 (6)	74.26 ± 0.7 (8)	76.15 ± 3.8 (4)	76.61 ± 3.21 (2)
OPM	74.18 ± 2.76 (5)	75.56 ± 4.2 (2)	69.17 ± 2.64 (11)	69.84 ± 6.69 (10)	73.04 ± 2.27 (7)	78.07 ± 3.7 (1)	70.46 ± 0.82 (9)	73.1 ± 2.33 (6)	72.53 ± 0.92 (8)	74.7 ± 4.14 (4)	75.23 ± 3.5 (3)
AvgRank	(5.13)	(4.5)	(8.88)	(10.13)	(7.5)	(1.88)	(8.25)	(6.75)	(7)	(3.63)	(2.38)

The experimental results on the rotating hyperplane dataset are summarized in Table V. It is observed that AWE gives the best accuracy, specificity, G-mean, precision, balanced accuracy, OPM, and thus has the best overall rank for rotating hyperplane dataset. However, EoE has the best sensitivity and F1-measure. Table VI gives the summary of empirical results on the rotating checkerboard with a constant drift rate dataset. It claims that BoostADWIN beats all other algorithms in all performance measurements on the rotating checkerboard with a constant drift rate dataset, whereas EoE stands fourth and AWE stands the last in the average ranking.

The performance measurement results on the weather dataset presented in Table VII show that though LevBag gives the best accuracy and precision, EoE has the best G-mean, F1-measure, balanced accuracy, and OPM. Hence it is the best performer on the weather dataset. The experimental results on the electricity pricing dataset are noted in Table VIII. It is observed that LevBag gives the best accuracy, G-mean, F1-measure, balanced accuracy, OPM, and thus has the best overall rank for the electricity pricing dataset. However, EoE has the best precision.

Table IX presents the experimental results on the airline's dataset. The results show that EoE has the best G-mean, F1-measure, and OPM, whereas WM gives the best accuracy, precision, and balanced accuracy. However, considering the overall average ranking WM and OAUE are the best and LevBag is the worst performer on the airline's dataset.

TABLE IX
AIRLINES DATASET EMPIRICAL RESULTS AND RANKING SUMMARY

	BagADWIN	BoostADWIN	DWM	ADACC	HT	LevBag	AWE	WM	AUE	OAUE	EoE
Acc	62.02 ± 1.42 (5)	58.2 ± 1.19 (9)	61.15 ± 2.11 (6)	56.13 ± 1.75 (11)	64.22 ± 0.3 (3)	57.96 ± 1.82 (10)	59.26 ± 0.92 (8)	64.55 ± 0.38 (1)	60.5 ± 2.25 (7)	64.46 ± 1.09 (2)	63.49 ± 1.63 (4)
Se	48.49 ± 1.88 (6)	56.1 ± 0.86 (1)	49.99 ± 2.43 (4)	53.34 ± 0.61 (2)	47.48 ± 2.1 (9)	46.98 ± 0.62 (10)	47.89 ± 0.28 (8)	48.4 ± 2.12 (7)	46.53 ± 1.84 (11)	48.81 ± 0.57 (5)	53.31 ± 0.73 (3)
Sp	72.89 ± 1.09 (4)	59.89 ± 1.85 (10)	70.12 ± 1.87 (7)	58.36 ± 2.7 (11)	77.67 ± 2.23 (1)	66.79 ± 2.79 (9)	68.39 ± 1.52 (8)	77.52 ± 2.38 (2)	71.71 ± 2.61 (5)	77.03 ± 1.54 (3)	71.67 ± 2.38 (6)
G-mean	59.45 ± 1.57 (5)	57.96 ± 1.1 (7)	59.2 ± 2.21 (6)	55.79 ± 1.57 (11)	60.69 ± 0.58 (4)	56.01 ± 1.52 (10)	57.23 ± 0.74 (9)	61.21 ± 0.51 (3)	57.77 ± 2.17 (8)	61.31 ± 0.95 (2)	61.81 ± 1.43 (1)
Prec	58.95 ± 1.85 (5)	52.93 ± 1.34 (10)	57.33 ± 2.7 (6)	50.76 ± 2.03 (11)	63.17 ± 1.54 (2)	53.27 ± 2.63 (9)	54.92 ± 1.35 (8)	63.46 ± 1.74 (1)	56.97 ± 3.32 (7)	63.08 ± 1.9 (3)	60.24 ± 2.49 (4)
F1	53.21 ± 1.88 (7)	54.46 ± 0.98 (4)	53.41 ± 2.55 (6)	52.01 ± 1.32 (8)	54.15 ± 0.96 (5)	49.91 ± 1.46 (11)	51.16 ± 0.69 (10)	54.86 ± 0.89 (3)	51.22 ± 2.44 (9)	55.03 ± 1.06 (2)	56.55 ± 1.47 (1)
Bal Acc	60.69 ± 1.47 (5)	57.99 ± 1.12 (9)	60.05 ± 2.14 (6)	55.85 ± 1.64 (11)	62.58 ± 0.08 (3)	56.88 ± 1.7 (10)	58.14 ± 0.85 (8)	62.96 ± 0.14 (1)	59.12 ± 2.21 (7)	62.92 ± 1.04 (2)	62.49 ± 1.54 (4)
OPM	58.23 ± 1.62 (5)	56.87 ± 1.09 (7)	57.92 ± 2.29 (6)	54.64 ± 1.55 (10)	59.69 ± 0.61 (4)	54.63 ± 1.6 (11)	55.88 ± 0.79 (9)	60.21 ± 0.59 (3)	56.49 ± 2.29 (8)	60.27 ± 1.04 (2)	60.62 ± 1.51 (1)
AvgRank	(5.25)	(7.13)	(5.88)	(9.38)	(3.88)	(10)	(8.5)	(2.63)	(7.75)	(2.63)	(3)

TABLE X
KDD CUP 10 PERCENT DATASET EMPIRICAL RESULTS AND RANKING SUMMARY

	BagADWIN	BoostADWIN	DWM	ADACC	HT	LevBag	AWE	WM	AUE	OAUE	EoE
Acc	99.95 ± 0.01 (4)	99.95 ± 0.01 (3)	99.16 ± 0.26 (8)	99.24 ± 0.26 (7)	99.93 ± 0.01 (6)	99.96 ± 0.01 (2)	97.3 ± 6.97 (11)	99.93 ± 0.01 (5)	97.69 ± 7.07 (10)	97.69 ± 7.04 (9)	99.97 ± 0 (1)
Se	99.86 ± 0.02 (4)	99.86 ± 0.05 (3)	98.2 ± 0.55 (7)	98.18 ± 0.62 (8)	99.79 ± 0.03 (6)	99.89 ± 0.03 (2)	88.6 ± 30.19 (11)	99.84 ± 0.03 (5)	90.07 ± 30.53 (10)	90.12 ± 30.42 (9)	99.91 ± 0.02 (1)
Sp	99.98 ± 0.01 (5)	99.98 ± 0 (4)	99.45 ± 0.17 (11)	99.56 ± 0.15 (10)	99.97 ± 0.01 (7)	99.98 ± 0 (2)	99.92 ± 0.03 (9)	99.96 ± 0.01 (8)	99.98 ± 0 (3)	99.98 ± 0.01 (6)	99.99 ± 0 (1)
G-mean	99.92 ± 0.01 (4)	99.92 ± 0.03 (3)	98.82 ± 0.36 (8)	98.87 ± 0.39 (7)	99.88 ± 0.02 (6)	99.94 ± 0.01 (2)	90.76 ± 26.14 (11)	99.9 ± 0.02 (5)	91.66 ± 25.92 (10)	91.76 ± 25.61 (9)	99.95 ± 0.01 (1)
Prec	99.92 ± 0.02 (4)	99.93 ± 0.01 (3)	98.17 ± 0.57 (11)	98.54 ± 0.51 (10)	99.92 ± 0.02 (5)	99.95 ± 0.01 (2)	99.72 ± 0.07 (8)	99.87 ± 0.03 (6)	99.7 ± 0.74 (9)	99.83 ± 0.26 (7)	99.96 ± 0.01 (1)
F1	99.89 ± 0.02 (4)	99.89 ± 0.03 (3)	98.19 ± 0.56 (8)	98.36 ± 0.56 (7)	99.85 ± 0.02 (6)	99.92 ± 0.01 (2)	89.55 ± 29.63 (11)	99.86 ± 0.02 (5)	90.47 ± 29.61 (10)	90.53 ± 29.39 (9)	99.93 ± 0.01 (1)
Bal Acc	99.92 ± 0.01 (4)	99.92 ± 0.03 (3)	98.82 ± 0.36 (8)	98.87 ± 0.38 (7)	99.88 ± 0.02 (6)	99.94 ± 0.01 (2)	94.26 ± 15.08 (11)	99.9 ± 0.02 (5)	95.03 ± 15.26 (10)	95.05 ± 15.21 (9)	99.95 ± 0.01 (1)
OPM	99.92 ± 0.01 (3.5)	99.92 ± 0.02 (3.5)	98.72 ± 0.39 (8)	98.82 ± 0.4 (7)	99.89 ± 0.02 (6)	99.94 ± 0.01 (2)	92.54 ± 20.91 (11)	99.9 ± 0.02 (5)	93.27 ± 20.87 (10)	93.33 ± 20.68 (9)	99.95 ± 0.01 (1)
AvgRank	(4.06)	(3.19)	(8.63)	(7.88)	(6)	(2)	(10.38)	(5.5)	(9)	(8.38)	(1)

TABLE XI

OPM RANKS SUMMARY OF ALL THE ALGORITHMS ON ALL THE DATASETS

	SEA	Agrawal	Rotating Hyperplane	RotChecker Board-CDR	Weather	Electricity	Airlines	KDD10 %	Avg	Final Rank
BagADWIN	4	2	5	6	9	5	5	3.5	4.94	4
BoostADWIN	11	9	11	1	3	2	7	3.5	5.94	5
DWM	8	10	3	8	6	11	6	8	7.5	8
ADACC	10	11	10	7	8	10	10	7	9.13	10
HT	6	7	9	10	11	7	4	6	7.5	8
LevBag	1	3	8	2	2	1	11	2	3.75	2
AWE	9	8	1	11	5	9	9	11	7.88	9
WM	7	6	4	9	10	6	3	5	6.25	6
AUE	5	5	7	5	7	8	8	10	6.88	7
OAUE	3	4	6	3	4	4	2	9	4.38	3
EoE	2	1	2	4	1	3	1	1	1.88	1

TABLE XII

IMAN-DAVENPORT TEST RESULTS (α=0.05) ON ALL PERFORMANCE METRICS

Metric	Acc	Se	Sp	G-mean	Prec	F1	BalAcc	OPM
p-value	0.000002	0.03613	0.03929	0.000408	0.002423	0.000077	0.000095	0.000035
Remark	Reject H_0	Reject H_0	Reject H_0	Reject H_0	Reject H_0	Reject H_0	Reject H_0	Reject H_0

TABLE XIII

PAIRWISE FRIEDMAN TEST RESULTS (α=0.05) ON ALL PERFORMANCE METRICS TO COMPARE EOE

	BagADWIN	BoostADWIN	DWM	ADACC	HT	LevBag	AWE	WM	AUE	OAUE
Acc	0.2	**0.0**	**0.0**	**0.0**	**0.0**	0.5	**0.0**	**0.0**	**0.0**	0.2
Se	0.1	0.1	**0.0**	**0.0**	**0.0**	0.1	**0.0**	**0.0**	**0.0**	**0.0**
Sp	0.8	0.1	0.8	0.1	0.3	0.7	0.4	0.7	0.8	0.8
G-mean	0.1	**0.0**	**0.0**	**0.0**	**0.0**	0.2	**0.0**	**0.0**	**0.0**	0.1
Prec	0.4	**0.0**	0.3	**0.0**	0.1	0.5	0.1	0.3	0.3	0.6
F1	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	0.2	**0.0**	**0.0**	**0.0**	0.1
Bal Acc	0.1	**0.0**	**0.0**	**0.0**	**0.0**	0.4	**0.0**	**0.0**	**0.0**	0.2
OPM	0.1	**0.0**	**0.0**	**0.0**	**0.0**	0.3	**0.0**	**0.0**	**0.0**	0.2

The empirical results on the KDD Cup 10 percent dataset are mentioned in Table X. It is observed that EoE outperforms in all performance metrics, and AWE is the worst performer on the KDD Cup 10 percent dataset.

We also used an overall performance and average rank as a compound evaluation metric. Table XI summarizes the OPM-based ranking of all algorithms on all datasets. Considering the average of all OPM ranks on all datasets, we defined the final rank of each algorithm. Fig. 2 depicts the average OPM ranks of all algorithms. It is observed that EoE with the least average OPM rank beats all other algorithms. It is the best performer, and ADACC is the worst performer based on OPM rank average.

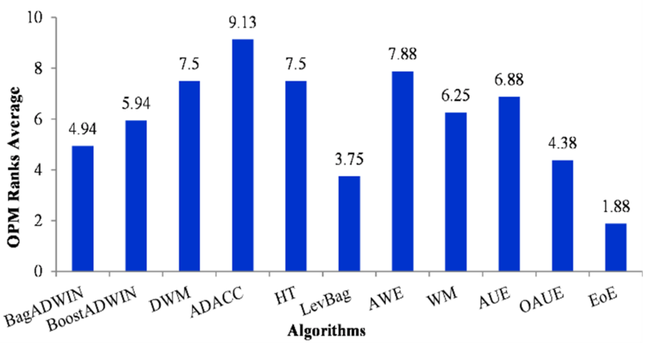

Fig. 2 Average OPM ranks of all algorithms

The EoE implements a bagging model of good performing sub ensembles with decision trees as base learners. With the integration of unstable base learners and the final prediction based on the plurality of the prediction results of sub ensembles, the EoE gives better performance than each of the sub ensembles. Also, the online learning and test-then-train approach employed by the EoE algorithm help to effectively handle streaming data and build an incremental model.

B. Statistical Results

To allow formal statistical distinctions among eleven algorithms over multiple data sets, we conduct the nonparametric statistical tests as recommended in García *et al.*[48]. Table XII provides the resultant p-values at a significant level α=0.05 of the Iman-Davenport test applied on all evaluation metrics of all the algorithms. Rejecting a null hypothesis (H_0: All algorithms show similar performance) underpins that at least one algorithm performs better than the other algorithms on all metrics and data sets. Since Table XI mentions EoE as the top-ranking algorithm, we apply the pairwise Friedman posthoc test with Li's correction to analyze whether EoE performs better among all algorithms on all metrics[48]. The results of the posthoc test at a significant level α=0.05 are recorded in Table XIII, with the bold-faced values indicating the significant performance increase of EoE compared to other state-of-the-art algorithms.

IV. CONCLUSION

The current communication addresses online ensemble learning in non-stationary data streams. These non-stationary data streams can have skewness in data samples and drifts in concepts. Numerous algorithms are proposed to deal with such non-stationary data. We have selected ten seminal ensemble learning approaches to build our ensembles to classify non-stationary binary data streams by reviewing the relevant literature. The proposed EoE bagging algorithm utilizes the learning capacity of each of these ten sub ensembles and results in the final prediction by the majority voting. Thus, mitigating limitations of an independent sub ensemble, the EoE gives better prediction results than that of an individual sub ensemble in classifying data streams with dynamicity.

In streaming data, as all the data samples are not arrived initially for training the model, the EoE provides an online learning model to classify data streams. To employ online learning in streaming data, we follow the test-then-train approach and Poisson parameter λ to approximate training samples. The Poisson approximation facilitates bootstrapping in streaming data samples.

The EoE and its state-of-the-art independent sub ensembles are empirically and statistically tested on a variety of figures of merits such as accuracy, sensitivity, specificity, G-mean, precision, F1-measure, balanced accuracy, and overall performance measure using different synthetic and real datasets. In the presented empirical study, none of the algorithms shows absolute superiority over all others. However, far more often, the proposed EoE algorithm outperforms other state-of-the-art algorithms with significance. The significant performance of EoE among the studied algorithms is verified through statistical tests like the Iman-Davenport test and the pairwise Friedman post hoc test with Li's correction.

Referring to the reported empirical work, we would like to work on some aspects in the future. Since this study demonstrates the empirical proof-of-concept of the EoE algorithm, we will explore its theoretical performance guarantees in the future. The performance gains of the EoE algorithm over the others do come at the cost of computational complexity. It is obvious because of sub ensembles built at each time step. We will experiment with different ensemble approaches to reduce computations. Also, adaptive online learning to effectively handle class imbalance and drifting concepts in non-stationary data streams will describe our future scope.

REFERENCES

[1] A. Boukhalfa, N. Hmina, and H. Chaoui, "Parallel processing using big data and machine learning techniques for intrusion detection," *IAES Int. J. Artif. Intell.*, vol. 9, no. 3, pp. 553–560, 2020, doi: 10.11591/ijai.v9.i3.pp553-560.

[2] E. De Luca, F. Fallucchi, R. Giuliano, G. Incarnato, and F. Mazzenga, "Analysing and visualizing tweets for U.S. president popularity," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 2, pp. 692–699, 2019, doi: 10.18517/ijaseit.9.2.8284.

[3] J. Sadhasivam and R. B. Kalivaradhan, "Sentiment analysis of Amazon products using ensemble machine learning algorithm," *Int. J. Math. Eng. Manag. Sci.*, vol. 4, no. 2, pp. 508–520, 2019, doi: 10.33889/ijmems.2019.4.2-041.

[4] P. D. Talagala, R. J. Hyndman, K. Smith-Miles, S. Kandanaarachchi, and M. A. Muñoz, "Anomaly Detection in Streaming Nonstationary Temporal Data," *J. Comput. Graph. Stat.*, vol. 29, no. 1, pp. 13–27, Jan. 2020, doi: 10.1080/10618600.2019.1617160.

[5] C.-C. Lin, D.-J. Deng, C.-H. Kuo, and L. Chen, "Concept drift detection and adaption in big imbalance industrial IoT data using an ensemble learning method of offline classifiers," *IEEE Access*, vol. 7, pp. 56198–56207, 2019, doi: 10.1109/ACCESS.2019.2912631.

[6] H. Zhang and Q. Liu, "Online learning method for drift and imbalance problem in client credit assessment," *Symmetry (Basel).*, vol. 11, no. 890, Jul. 2019, doi: 10.3390/sym11070890.

[7] 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," *Inf. Fusion*, vol. 54, pp. 128–144, 2020, doi: 10.1016/j.inffus.2019.07.006.

[8] M. Bahri, A. Bifet, J. Gama, H. M. Gomes, and S. Maniu, "Data stream analysis: Foundations, major tasks and tools," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 11, no. 3, p. e1405, May 2021, doi: 10.1002/widm.1405.

[9] 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.

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

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

[12] I. Khamassi, M. Sayed-Mouchaweh, M. Hammami, and K. Ghédira, "Discussion and review on evolving data streams and concept drift adapting," *Evol. Syst.*, vol. 9, pp. 1–23, 2018, doi: 10.1007/s12530-016-9168-2.

[13] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363, Dec. 2019, doi: 10.1109/TKDE.2018.2876857.

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

[15] 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, doi: 10.1016/j.inffus.2017.02.004.

[16] R. Jani, N. Bhatt, and C. Shah, "A survey on issues of data stream mining in classification," in *Information and Communication Technology for Intelligent Systems (ICTIS 2017) - Volume 1, Smart Innovation, Systems and Technologies*, vol. 83, S. C. Satapathy and A. Joshi, Eds. Ahmedabad, India: Springer, Cham, 2018, pp. 137–143.

[17] Janardan and S. Mehta, "Concept drift in streaming data classification: Algorithms, platforms and issues," *Procedia Comput. Sci.*, vol. 122, pp. 804–811, 2017, doi: 10.1016/j.procs.2017.11.440.

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

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

[20] M. M. Idrees, L. L. Minku, F. Stahl, and A. Badii, "A heterogeneous online learning ensemble for non-stationary environments," *Knowledge-Based Syst.*, vol. 188, p. 104983, Jan. 2020, doi: 10.1016/j.knosys.2019.104983.

[21] N. Littlestone and M. K. Warmuth, "The Weighted Majority algorithm," *Inf. Comput.*, vol. 108, no. 2, pp. 212–261, 1994, doi: 10.1006/inco.1994.1009.

[22] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, 2003, pp. 226–235, doi: https://doi.org/10.1145/956750.956778.

[23] J. Kolter and M. Maloof, "Dynamic Weighted Majority : An ensemble method for drifting concepts," *J. Mach. Learn. Res.*, vol. 8, pp. 2755–2790, 2007, doi: 10.1.1.140.2481.

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

[25] S. Ren, B. Liao, W. Zhu, Z. Li, W. Liu, and K. Li, "The Gradual Resampling Ensemble for mining imbalanced data streams with concept drift," *Neurocomputing*, vol. 286, pp. 150–166, 2018, doi: 10.1016/j.neucom.2018.01.063.

[26] 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, pp. 139–148, doi: https://doi.org/10.1145/1557019.1557041.

[27] H. M. Gomes *et al.*, "Adaptive random forests for evolving data stream

classification," *Mach. Learn.*, 2017, doi: 10.1007/s10994-017-5642-8.

[28] R. S. M. de Barros, J. I. G. Hidalgo, and D. R. de L. Cabral, "Wilcoxon Rank Sum Test Drift Detector," *Neurocomputing*, vol. 275, pp. 1954–1963, 2018, doi: 10.1016/j.neucom.2017.10.051.

[29] A. Cano and B. Krawczyk, "Kappa Updated Ensemble for drifting data stream mining," *Mach. Learn.*, vol. 109, no. 1, pp. 175–218, Jan. 2020, doi: 10.1007/s10994-019-05840-z.

[30] V. Losing, B. Hammer, and H. Wersing, "Tackling heterogeneous concept drift with the Self-Adjusting Memory (SAM)," *Knowl. Inf. Syst.*, vol. 54, pp. 171–201, 2018, doi: 10.1007/s10115-017-1137-y.

[31] S. Priya and R. A. Uthra, "Comprehensive analysis for class imbalance data with concept drift using ensemble based classification," *J. Ambient Intell. Humaniz. Comput.*, 2020, doi: 10.1007/s12652-020-01934-y.

[32] H. Hu, M. Kantardzic, and T. S. Sethi, "No Free Lunch Theorem for concept drift detection in streaming data classification: A review," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 10: e1327, no. 2, Mar. 2020, doi: 10.1002/widm.1327.

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

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

[35] D. Brzezinski and J. Stefanowski, "Combining block-based and online methods in learning ensembles from concept drifting data streams," *Inf. Sci. (Ny).*, vol. 265, pp. 50–67, 2014, doi: 10.1016/j.ins.2013.12.011.

[36] R. S. M. Barros, D. R. L. Cabral, P. M. Gonçalves, and S. G. T. C. Santos, "RDDM: Reactive Drift Detection Method," *Expert Syst. Appl.*, vol. 90, pp. 344–355, 2017, doi: 10.1016/j.eswa.2017.08.023.

[37] S. Ren, B. Liao, W. Zhu, and K. Li, "Knowledge-Maximized Ensemble algorithm for different types of concept drift," *Inf. Sci. (Ny).*, vol. 430–431, pp. 261–281, 2018, doi: 10.1016/j.ins.2017.11.046.

[38] P. Sidhu and M. P. S. Bhatia, "A novel online ensemble approach to handle concept drifting data streams: diversified dynamic weighted majority," *Int. J. Mach. Learn. Cybern.*, vol. 9, pp. 37–61, 2018, doi: 10.1007/s13042-015-0333-x.

[39] Z. Yang, S. Al-Dahidi, P. Baraldi, E. Zio, and L. Montelatici, "A novel concept drift detection method for incremental learning in non-stationary environments," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 1, pp. 309–320, Jan. 2020, doi: 10.1109/TNNLS.2019.2900956.

[40] Y. Lu, Y.-M. Cheung, and Y. Y. Tang, "Adaptive Chunk-based Dynamic Weighted Majority for imbalanced data streams with concept drift," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 8, pp. 2764–2778, 2020, doi: 10.1109/TNNLS.2019.2951814.

[41] Z. Li, W. Huang, Y. Xiong, S. Ren, and T. Zhu, "Incremental learning imbalanced data streams with concept drift: The dynamic updated ensemble algorithm," *Knowledge-Based Syst.*, vol. 195, no. 105694, 2020, doi: 10.1016/j.knosys.2020.105694.

[42] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 7th SIAM International Conference on Data Mining*, 2007, pp. 443–448, doi: 10.1137/1.9781611972771.42.

[43] G. Jaber, A. Cornuejols, and P. Tarroux, "A new online 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, 2013, pp. 595–604.

[44] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'00)*, 2000, pp. 71–80, doi: 10.1145/347090.347107.

[45] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," in *Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2010, Lecture Notes in Computer Science*, vol. 6321, J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds. Barcelona, Spain: Springer, Berlin, Heidelberg, 2010, pp. 135–150.

[46] D. Brzezinski and J. Stefanowski, "Reacting to different types of concept drift: The accuracy updated ensemble algorithm," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 1, pp. 81–94, 2014, doi: 10.1109/TNNLS.2013.2251352.

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

[48] 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," *Inf. Sci. (Ny).*, vol. 180, no. 10, pp. 2044–2064, 2010, doi: 10.1016/j.ins.2009.12.010.