Journal of Advanced Computer Science & Technology, 13 (1) (2026) 44-65

SPC

Journal of Advanced Computer Science & Technology

Website: www.sciencepubco.com/index.php/JACST
https://doi.org/10.14419/nse20c73
Research paper

Concept Drift Detection Method Based on Shifting Classifier's Mean Error Rate

Nagaraj Honnikoll, Ishwar Baidari *

Department of Computer Science, Karnatak University, Dharwad 580001, Karnataka, India
**Corresponding author E-mail: ishwarbaidari@kud.ac.in*

Received: 08-05-2026, Accepted: 08-08-2026, Published: 21-08-2026

Abstract

Concept drift detection is an important technique for handling issues in real-time online learning. Learning in a real-time environment is challenging because data distribution is dynamic. This dynamic nature of data distribution affects the performance of the learning algorithm. Therefore, drift detection algorithms are used to handle such issues. In this article, we introduce a simple algorithm, the Shifting Classifier's Mean Error Rate (SCMER), which identifies the drift position using the classifier's mean error rate, computed from a range of instances (e.g., from 1 to n). The algorithm identifies the drift position when the deviation between two cumulative mean error rates is greater than a threshold value, ϵ. We proposed a Hellinger distance-inspired metric for measuring the deviation between two cumulative mean error rates. For hypothesis testing, we adapted the Hoeffding Bound. Experimental analysis shows that our algorithm detects drifts with shorter detection delays, fewer false positives, and fewer false negatives when compared to state-of-the-art methods.

Keywords: *Concept Drift; Hellinger Distance; Data Stream; Online Learning; Hoeffding Bound.*

1. Introduction

Learning from data in continuous evolving environment is very difficult. This challenge depends not only on the speed and size of the incoming data but also on distributional changes. These changes can negatively affect a classifier's performance. Therefore, we need algorithms equipped with drift detection techniques to identify such changes and take appropriate measures to adapt the algorithm accordingly.[1]

Employing drift detection methods leads to a decrease in false positives and false negatives. Such algorithms are inherently fast, designed to detect the drifts as soon as they occur. When a detection algorithm produces a high number of false positives, the classifier needs to be retrained frequently. This frequent training requires more resources, which can overwhelm the system. On the other hand, a high number of false negatives downgrades classification accuracy, as the algorithm fails to identify actual drift positions. Such miscalculations are very costly. Accurately approximating the drift position and identifying it quickly allows detectors to utilize the data effectively. Concept drift detectors are used in many online learning applications, such as monitoring sensor data. [2], spam filtering of e-mail messages[3], sentiment analysis [4], predicting equipment failure, and many more [5],[6].

Many state-of-the-art methods are unable to detect drifts when a concept remains in the same state for a long period. For example, to detect a drift, DDM calculates the mean error rate, which requires a large number of prediction errors. The concept can remain at a warning level for a huge number of instances before finally reaching the drift level. Such behavior increases the memory usage of the classifier, which is learning in parallel. These obstacles prevent many algorithms from detecting drifts quickly. Consequently, the base learner may start slowly adapting to the new concept without formally detecting the existing drift.

This article proposes a novel method based on a shifting classifier's mean error rate. The technique involves the algorithm continuously processing a range of (1 to n) instances and shifting the classifier's mean error rate for every new segment of (1 to n) instances. This shifting helps the algorithm avoid remaining in the same state. To process a range of instances, many state-of-the-art algorithms use a windowing method, which consumes significant memory. In contrast, the proposed algorithm does not use a window method, leading to less memory consumption. We used a Hellinger distance-inspired metric to measure the difference between two cumulative mean error rates and have adapted the Hoeffding Bound for hypothesis testing.

The experimental results show that the proposed algorithm improves global accuracy. We used MOA [7]for the experimental evaluation of our proposed algorithm against other detectors, utilizing both artificial and real-world datasets.

Table 1 refers to the notations used with their meanings. The rest of the article is organized as follows: Section 2 presents a survey of related work; Section 3 describes the proposed algorithm in detail; Section 4 presents the configuration and datasets used for the experiments; Section 5 presents the experimental results, evaluation, and statistics for comparing accuracies and analyzing drift identifications; and finally, Section 6 draws conclusions and discusses future work.

Table 1: Used Notations and Their Meanings

Notations	Meaning
P_{min}	Stores the minimum observed cumulative mean-error rate value seen in the data stream.
P_{cur}	Stores the cumulative mean-error rate value of the current n instances.
P_{next}	Stores the cumulative mean-error rate value of the next n instances.
Pred	Prediction result of the classifier, either true (1) or false (0).
$d(P_{cur}, P_{min})$	Hellinger Distance inspired metric calculated between P_{cur} and P_{min} using equation 5.
ϵ	Hoeffding Bound value calculated from equation 6.
H_0	Null Hypothesis (No drift detected in the data stream).
H_1	Alternative Hypothesis (Drift detected in the data stream).

2. Related Work

Several methods have been proposed to detect concept drifts in data streams, including DDM. [8], EDDM [9], ADWIN [10], STEPD [11], PHT [12], PL [13], ECDD [14], HDDM [15], FHDDM [16], RDDM [17] and VFDDM [18].

Drift Detection Method (DDM) [8]is a widely-known drift detector, that identifies drifts by monitoring a classifier's error rate and its corresponding standard deviation. For every data point i in the stream, DDM records the error rate p_i (the probability of wrong prediction) and calculates its standard deviation as ($s_i = \sqrt{p_i \times (1 - p_i)/i}$). Based on the Probably Approximately Correct (PAC) learning model [19], the authors state that error rate p_i should decrease as the number of examples i increases, if the data distribution is stationary. Therefore, an increase in error rate suggests a change in the data distribution, indicating that the classifier is no longer performing optimally and should be retrained for better accuracy.

Hoeffding-based Drift Detection Method (HDDM) [15] comprises a group of methods that monitor the mean performance of the base learner over time to identify significant changes. Unlike DDM, which assumes a Bernoulli distribution, HDDM makes no assumptions about the distribution of the incoming data. Two versions of HDDM are introduced: $HDDM_{A-Test}$, which uses a moving average and is more suitable for detecting abrupt concept drifts, and $HDDM_{W-Test}$, which uses a weighted moving average and is better for detecting gradual drifts.

In Statistical Test of Equal Proportions (STEPD) [11] algorithm computes the difference between two windows of data (a recent window and an older one). If a significant difference is found, it confirms a change in the distribution. The standard parameters for STEPD are a window size of ($w = 30$). and a confidence level for drift detection of ($\alpha_d = 0.003$) and a confidence level for warnings of ($\alpha_w = 0.05$).

Exponentially weighted moving average charts for detecting concept drift (ECDD) [14] is adapted from the Exponentially Weighted Moving Average (EWMA) [20] technique. While traditional EWMA requires prior knowledge of the mean and standard deviation, ECDD does not. ECDD is based on three parameters, although only two are required for its implementation in MOA. Weights play a crucial role in ECDD by differentiating recent instances from old ones, and as such, no default values are needed.

The Fast Hoeffding Drift Detection Method (FHDDM) [16] uses Hoeffding's inequality and a sliding window technique to detect drifts. FHDDM identifies a drift when a significant difference is observed between the maximum and the most recent probabilities of the classifier's correct predictions. This difference is measured by a $\delta(P)$ metric, and a drift is declared if $\delta(P) \geq \epsilon$, where ϵ is derived from Hoeffding's inequality.

Reactive Drift Detection Method (RDDM[17] is an improvement upon DDM. Among other modifications, its strategy involves discarding older instances from very long concepts to identify drifts earlier and improve final accuracy, addressing DDM's difficulties with lengthy concepts. These enhancements are particularly important for long concepts involving gradual drifts.

Concept drift detection based on Fisher's Exact test [21] is inspired by the Statistical Test of Equal Proportions (STEPD) [11]. This method was proposed to overcome STEPD's inefficiency with small, imbalanced, and/or sparse data samples. Three methods using Fisher's Exact Test—FTDD, FSDD, and FPDD—were introduced to address these deficiencies.

Wilcoxon Rank Sum Test Drift Detector (WSTD) [22] is another method motivated by STEPD [11]. The authors effectively implemented the Wilcoxon rank sum statistical test to detect concept drifts. WSTD overcomes some of STEPD's deficiencies and improves accuracy in most scenarios. Although WSTD provides powerful overall performance, it works particularly well with large datasets involving abrupt concept drifts while maintaining high detection precision.

Variance Feedback Drift Detection Method for Evolving Data Streams Mining (VFDDM) [18] proposed a framework for the detection of concept drifts based on variance. This framework can utilize various drift tests for detection of drift in data stream. The workflow consists of two stages (i) shift variance estimation and (ii) variance feedback strategy. Shift variance estimation establishes baseline variance which is utilized in variance feedback strategy. In variance feedback strategy, based on the estimated variance for the dynamic adjustment of weighting for the current window is performed for better detection of drift occurrence. Based on the framework, three distinct drift detectors are provided: VFDDM_H, VFDDM_M and VFDDM_K.

Adaptive windowing-based concept drift detection and adaptation (ADA) [23] framework proposed to support Human-to-machine applications, which employs dynamic changes to the sliding window sizes based on the drift detection results. For detection techniques it employed the Hoeffding's inequality-based algorithm for effectively detecting incremental H2M application traffic drift in a dynamic network. They showed that ADA improved uplink latency reduction and it responded rapidly and adopted to changing H2M applications and traffic load scenarios.

Dynamic similarity weighted evolving fuzzy system (DSW-EFS) [24] as an ensemble system for concept drift in data streams by assigning weights to base learners based on the similarity between data distributions, and each base learner describes only one data distribution in the data stream. This ensemble method enables DSW-EFS to describe multiple data distributions in the data stream and quickly adapt to multiple concept drifts. To ensure the accuracy of drift detection, we propose a Gaussian mixture model (GMM)-based concept drift detection algorithm that can obtain the similarity between data distributions. This detection method can achieve high accuracy based on a smaller sliding window size.

Concept Drift Detection Based on Deep Neural Network Combined with Autoencoder (DNN+AE-DD) [25] is a deep neural network is first employed as the base model for pretraining, and the hidden layer parameters of the model are transferred to a network with an identical structure for stream data processing, where certain hidden layers are frozen. Subsequently, the hidden layer outputs from both the pretraining and stream data processing phases are collected and used as training and testing data to initialize and predict using an autoencoder model. Concept drift is then detected by combining the reconstruction error of the autoencoder with the 3σ principle.

3. Proposed Work

We present our classifier's mean error shifting method, which uses the Hellinger distance-inspired metric and the Hoeffding's inequality to detect the location of drifts in data streams. Unlike methods based on a sliding window strategy, the proposed approach continuously processes a range of (1 - n) instances from the classification predictions, recording a 1 for true predictions and a 0 for false predictions. As each new instance in the range is processed, the algorithm computes the cumulative mean error rate for the variables P_{cur} and P_{next} using the following equations:

$$P_{cur} = P_{cur} + (Pred - P_{cur})/P_{curIndex} \tag{1}$$

$$P_{next} = P_{next} + (Pred - P_{next})/P_{nextIndex} \tag{2}$$

Subsequently, the cumulative mean error rate is shifted by assigning the value of P_{next} to P_{cur}. This shifting operation occurs for every range of (1 to n) instances. This process avoids explicit usage of sliding window storage and operates with constant memory complexity. An example of this process is illustrated in Figure 1.
The value for P_{min} is updated using Equation 3 if the current value of P_{cur} is less than the existing P_{min}

$$P_{min} = P_{cur}\ ,\ \text{if}(P_{cur} \leq P_{min}) \tag{3}$$

According to the Probably Approximately Correct (PAC) learning model, classification accuracy should increase or stabilize as the number of instances grows; otherwise, the likelihood of a concept drift increases. Therefore, the value of P_{cur} is expected to decrease or remain stable as more instances are processed. To detect a significant change indicating drift, we proposed a Hellinger distance inspired metric to measure the difference between $P_{cur}$and P_{min}, as defined in Equation 5, which is derived as follows:

Definition 1: *Hellinger Distance*

For probability distributions $P = p_1, p_2, \ldots, p_n$, $Q = q_1, q_2, \ldots, q_n$supported on $[n]$, the Hellinger distance between them is defined as

$$h(P, Q) = \frac{1}{\sqrt{2}} \times \sqrt{\sum_{i=1}^{n} \left(\sqrt{p_i} - \sqrt{q_i}\right)^2}$$

As in the data stream, the classifier predications are produced which is in the form of binary which can be modelled as Bernoulli random variables. For Bernoulli distributions with parameter (P) and (Q), the Hellinger distance is defined as

$$h^2(P, Q) = 1 - \left(\sqrt{pq} + \sqrt{(1-p)(1-q)}\right) \tag{4}$$

To reduce computational complexity for data stream for detection of drift, we proposed metric which computes the deviation between two cumulative mean error rates (P_{cur} and P_{min}) over the range (1 to n). The proposed metric is inspired form Hellinger distance metric given in equation 4 which is calculated as follows,

$$\triangle P = d(P_{cur}, P_{min}) = \frac{1}{\sqrt{2}} \times \sqrt{P_{cur} - P_{min}}, \qquad P_{cur} \geq P_{min}. \tag{5}$$

The proposed metric satisfies the following metric properties,

- $\triangle$ Pis strictly bounded between 0 and 1.
- $\triangle P = 0$, if two cummulative mean error rates are identical ($P_{cur} = P_{min}$).
- $\triangle P = 1$, if two cummulative mean error rates are completely disjoint (e.g., $P_{cur} = 1$, and $P_{min} = 0$).
- For fixed P_{min}, $\triangle$ P will monotonically increases for increasing P_{cur}.

These properties follow properties of Hellinger distance metric. Based on these properties, the proposed Hellinger distance inspired metric is used for the detection of drifts in data stream.
Since, the proposed metric produces the deviations between the two cumulative mean error rates, which integrates naturally with Hoeffiding inequality for hypothesis testing. Hence, we used Hoeffding inequality for hypothesis testing which is given for n independent observations, the true mean and the estimated mean will not differ by more than ε with probability $1 - \delta$, where:

$$\varepsilon = \sqrt{\frac{1}{2n} \log \frac{2}{\delta}} \tag{6}$$

Where $\delta \in (0,1)$is a user defined confidence parameter.

Based on the above Hellinger distance inspired metric and Hoeffding inequality, the hypothesis test for drift detection in a data stream is formulated as follows:

$$\begin{cases} H_0, & \text{if } \triangle P \leq \varepsilon \\ H_1, & \text{if } \triangle P > \varepsilon. \end{cases}$$

H_0 is the null hypothesis, indicating no drift, while H_1 is the alternative hypothesis, confirming a drift. If H_1 is accepted, the algorithm resets its statistics. The complete proposed approach is described in Algorithm 1.

Initial Values: $n = 5, \delta = 0.2 \rightarrow \epsilon = 0.4798, P_{cur} = 0, P_{next} = 0, P_{min} = Inf$

Instance	*Prediction*	*Index* P_{cur}	P_{cur}	*Index* P_{next}	P_{next}	P_{min}	ΔP	
1	0	1	0.0000	1	0.0000	Inf	?	
2	1	2	0.5000	2	0.0000	Inf	?	
3	1	3	0.6667	3	0.0000	Inf	?	
4	0	4	0.5000	4	0.0000	Inf	?	
5	0	5	0.4000	5	0.0000	Inf	?	
6	0	6	0.3333	1	0.0000	0.3333	0.0000	
7	0	7	0.2857	2	0.0000	0.2857	0.0000	
8	0	8	0.2500	3	0.0000	0.2500	0.0000	
9	1	9	0.3333	4	0.2500	0.2500	0.2041	
10	1	10	0.4000	5	0.4000	0.2500	0.2739	
10	**1**	**5**	**0.4000**	**5**	**0.4000**	**0.2500**	**0.2739**	**Shifting Classifier's Mean Error Rate**
11	0	6	0.3333	1	0.0000	0.2500	0.2041	
12	**1**	**7**	**0.4286**	**2**	**0.5000**	**0.2500**	**0.2988**	**True Location of Drift**
13	1	8	0.5000	3	0.6667	0.2500	0.3536	
14	0	9	0.4444	4	0.5000	0.2500	0.3118	
15	1	10	0.5000	5	0.6000	0.2500	0.3536	
15	**1**	**5**	**0.6000**	**5**	**0.6000**	**0.2500**	**0.4183**	**Shifting Classifier's Mean Error Rate**
16	1	6	0.6667	1	1.0000	0.2500	0.4564	
17	**1**	**7**	**0.7143**	**2**	**1.0000**	**0.2500**	**0.4818**	**Detected Location of Drift**

Fig. 1: Illustration of Shifting Classifier's Mean Error Rate Used in SCMER.

Figure 1 provides an illustrative example of the proposed algorithm's operation. In this example, the parameters n and δ are set to 5 and 0.2, respectively. Using Equation 5, the value of ε is calculated to be 0.4798. An actual drift occurs in the data stream immediately after the 11^{th} instance. For the first five instances, the values of P_{next} and P_{min} are null and zero. At the 6^{th} instance, the P_{cur} becomes 0.3333, and since this is the first computed value, P_{min} is also set to 0.3333. From 6^{th} to 10^{th} instance, the cumulative mean error is continuously computed using Equations 1 and 2, and the value of P_{min} is updated whenever a new minimum is encountered.

Algorithm 1 SCMER Algorithm

procedure SCMER (stream, n , δ)
Initialize $P_{curIndex} \leftarrow 1, P_{cur} \leftarrow 1, P_{min} \leftarrow 1, P_{next} \leftarrow 1, P_{nextIndex} \leftarrow 1$,
$\epsilon \leftarrow \sqrt{\left(\frac{1}{2\times n}\right) \times \ln\left(\frac{2}{\delta}\right)}$ ▻ Hoeffding Bound
for instance, in stream do
pred ← prediction(instance)
$P_{cur} \leftarrow P_{cur} + (pred - P_{cur})/P_{curIndex}$ ▻ Calculation of Cummulative Mean Error Rate
$P_{curIndex} \leftarrow P_{curIndex} + 1$
if $P_{curIndex} >= n$ then
if $((P_{cur} < P_{min})$then
$P_{min} \leftarrow p_{cur}$ ▻ Storing P_{min} value seen so far
end if
$P_{next} = P_{next} + (pred - P_{next})/P_{nextIndex}$
$P_{nextIndex} \leftarrow P_{nextIndex} + 1$
if $P_{curIndex} == 2 \times n$ then ▻ Proposed Shifting Classifier's Mean Error Rate starts here
$P_{cur} \leftarrow P_{next}$
$P_{curIndex} \leftarrow P_{nextIndex}$
$P_{next} \leftarrow 1$
$P_{nextIndex} \leftarrow 1$
end if
$\Delta P \leftarrow \frac{1}{\sqrt{2}} \times \sqrt{P_{cur} - P_{min}}$ ▻ Hellinger Distance Inspired Metric Calculation
if $(\Delta P >= \epsilon)$ then ▻ Testing for drift occurance.
changeDetected ← true
else
changeDetected ← false
end if
if changeDetected = true then
reset $P_{curIndex}, P_{cur}, P_{min}, P_{next}, P_{nextIndex}$ ▻ Reset the statistics as the classifier is being reset
end if
end if
end for
end procedure

A key step occurs at the 15^{th} instance, where the algorithm updates its statistics by shifting the value of P_{next}(which is 0.6000 in this case) to P_{cur}. This process is repeated for every subsequent range of (1 to n) instances. At 17^{th}position the difference between $P_{cur}$and P_{min}becomes more than the value of ε. In this case, the algorithm locates the drift.

3.1. Computational complexity analysis

The proposed approach maintains a fixed number of scalar statistics: P_{cur}, P_{next}, P_{min}, P_{count} and $P_{nextcount}$. Because these values are independent of the data stream length (N), SCMER has a constant memory complexity of $O(1)$. In contrast, approaches such as FHDDM, STEPD, ADWIN, and RDDM store recent observations in a window of size w, resulting in a memory complexity of $O(w)$. For each incoming instance, SCMER performs a constant number of arithmetic operations to update its scalar statistics. Therefore, its per-instance runtime complexity is $O(1)$, and its total runtime complexity for a stream of N instances are $O(N)$. By eliminating window maintenance, SCMER reduces memory overhead and per-instance computational cost, making it suitable for data stream environments.

4. Experimental Setting

This section outlines the experimental setup used to test and evaluate our proposed method alongside other benchmark methods. We employed Naive Bayes and Hoeffding Tree classifiers as the base learners, as these models are widely used in data stream research.
For the evaluation, we selected three artificial data stream generators (Agrawal [26], Asset Negotiation [27], and RBF [26]) which incorporate both gradual and abrupt concept drifts. These generators were used to create datasets of varying sizes and concept distributions. Additionally, eight real-world datasets were included in the evaluation. Table 2 summarizes the key characteristics of all datasets used.
To evaluate accuracy, we used the prequential evaluation method [28].

Table 2: Characteristics of Artificial Generated Datasets and Real-World Datasets

Dataset	#Instances	#Attributes	#Class	Noise	#Drifts	Drift Type
AGRAWAL	50K, 100K, 300K, 500K	9	2	10 %	4	Abrupt, Gradual
RBF	50K, 100K, 300K, 500K	4	2	0 %	4	Abrupt, Gradual
Asset Negotiation	50K, 100K, 300K, 500K	5	2	0.05 %	4	Abrupt, Gradual
Airlines [29]	539,383	24	4	0	Unknown	Unknown
Poker hand [15], [30]	829,201	10	10	0	Unknown	Unknown
Cover type [15]	581,012	53	7	0	Unknown	Unknown
Electricity [8], [31]	45,312	9	2	0	Unknown	Unknown
Online Shoppers [31]	12,330	18	2	0	Unknown	Unknown
Codrna [32]	488,565	8	2	0	Unknown	Unknown
Avila [33]	20,867	10	12	0	Unknown	Unknown
Adult [34]	100,000	14	2	0	Unknown	Unknown

4.1. Parameter sensitivity

The SCMER algorithm has two principal parameters which are the minimum shifting interval (n) and the Hoffding confidence parameter (δ). The parameter (n) controls the number of instances which will be used to calculate the cumulative mean error rates. If smaller values of (n) makes faster adaptation to drifts changes but may increase sensitivity to noise, leading to higher false positive rates. Conversely, if large values of (n) provides smoother mean error rates estimations, which makes stable drift detections at the cost of increased in detection delays of drifts in data stream. The parameter (δ) influences the Hoeffding threshold (ϵ). Smaller value of (δ) increases the threshold (ϵ), resulting in fewer false alarms and more conservative drift detections. Also, larger value of (δ) reduces the threshold (ϵ), increasing the more number of false positives. Table 3 provides the analysis of SCMER's parameter values for Agrawal 100K dataset.

Table 3: Parameter Sensitivity Analysis of SCMER on Agarwal 100K Dataset

N	δ	μdist	FP	FN	Accuracy
	10^{-2}	7.5	303	0	75.38
10	10^{-4}	106.5	2	2	85.26
	10^{-6}	-	-	-	74.22
	10^{-2}	8.5	341	0	74.90
30	10^{-4}	13	50	0	79.42
	10^{-6}	28.5	0	0	86.92
	10^{-2}	51	259	0	75.39
50	10^{-4}	12	101	0	76.90
	10^{-6}	22	2	0	85.70
	10^{-2}	11.5	172	0	75.76
100	10^{-4}	9.5	84	0	77.31
	10^{-6}	48	31	0	79.83

For the experimental settings, the parameters for our method, SCMER, were set as follows: the minimum number of instances after which the shifting of the classifier's mean error rate begins is $n = 30$, which is standard in most drift detectors, and the confidence parameter for calculating the Hoeffding Bound is $\delta = 10^{-6}$ which is set based on the FHDDM approach. These values were selected because they are commonly adopted configuration values in most of the prior drift detection methods. Also, these values provided a stable balance between detection responsiveness and fewer false positives. The paramters used for the compared models is provided in the Table 4.

5. Experimental Results and Analysis

This section presents an analysis of the results obtained from experiments conducted on the 32 tested dataset configurations. We discuss the accuracy results, concept drift identification, runtime and memory analysis separately.

Table 4: Algorithm Parameter Settings for the Experiments

Algorithm	Parameters
DDM	$n = 30, w = 2.0, d = 3.0$
ECDD	$n = 30, l = 0.2$
STEPD	$r = 30, o = 0.003, w = 0.05$
FHDDM	$s = 25, d = 10^{-6}$
RDDM	$n = 129, w = 1.773, o = 2.258, x = 40000, y = 7000, z = 1400$
HDDM_A	$d = 0.01, w = 0.005, t = \text{Two} - \text{sided}$
HDDM_W	$d = 0.01, w = 0.005, t = \text{One} - \text{sided}, m = 0.05$
VFDD	$s = 50, v = 50, d = 0.01, c = 10^{-8}$

5.1. Accuracy results and analysis

This section presents the accuracy results obtained from the tested drift detection methods. Table 5 and Table 6 shows the accuracy results for DDM, ECDD, STEPD, FHDDM, RDDM, HDDM_A, HDDM_W, VFDD (For the experiment we used VFDDM_H [18] which is termed as VFDD) and our proposed method, SCMER. For each dataset, the highest accuracy achieved is indicated in bold.

Table 5: Average Accuracies in Percentage (%) Using Naive Bayes, with 95\% Confidence Intervals in the Artificial Datasets

TYPE-SIZE	DATASET	DDM	ECDD	STEPD	FHDDM	HDDM_A_Test
	Agrawal	77.41±4.58	76.72±5.36	78.58±4.99	78.65±5.05	78.75±5.00
Abrupt-50K	RBF	61.69±4.11	58.91±3.76	60.34±5.41	61.35±4.78	61.86±4.16
	Asset Negotiation	88.28±1.81	87.65±1.79	88.72±1.98	88.79±2.04	88.74±2.02
	Agrawal	77.51±4.69	76.38±5.11	78.31±5.06	78.43±5.22	78.42±4.97
Abrupt-100K	RBF	61.39±3.90	59.23±3.67	60.60±4.90	61.06±4.51	61.61±3.99
	Asset Negotiation	88.46±1.92	88.09±1.84	89.13±2.03	88.86±2.08	88.80±2.04
	Agrawal	76.78±4.98	75.83±5.36	77.53±5.48	77.67±5.60	77.96±5.41
Abrupt-300K	RBF	61.05±4.27	59.57±3.98	60.44±4.69	60.33±4.65	61.13±4.35
	Asset Negotiation	88.05±1.72	87.74±1.72	88.77±1.88	88.46±2.01	88.44±2.00
	Agrawal	77.27±4.79	76.56±5.24	78.14±5.29	78.24±5.40	78.67±5.28
Abrupt-500K	RBF	61.04±3.86	60.12±3.86	61.07±4.46	60.92±4.51	61.86±4.22
	Asset Negotiation	87.99±1.76	87.77±1.65	88.82±1.81	88.37±1.99	88.37±1.99
	Agrawal	77.26±4.47	76.60±5.31	78.36±4.88	78.56±5.01	78.72±4.97
Gradual-50K	RBF	61.17±3.92	58.97±3.75	59.63±5.17	60.49±4.48	61.50±4.03
	Asset Negotiation	88.24±1.85	87.74±1.72	88.70±2.04	88.73±2.06	88.71±2.06
	Agrawal	77.51±4.69	76.39±5.11	78.18±5.01	78.38±5.19	78.39±4.95
Gradual-100K	RBF	60.94±3.70	59.19±3.65	60.24±4.80	60.64±4.39	61.06±3.80
	Asset Negotiation	88.47±1.95	87.98±1.84	89.00±2.06	88.83±2.09	88.79±2.06
	Agrawal	76.78±4.98	75.82±5.36	77.52±5.47	77.65±5.59	77.95±5.41
Gradual-300K	RBF	60.79±4.19	58.94±3.82	60.38±4.66	60.21±4.61	60.97±4.30
	Asset Negotiation	88.17±1.79	87.82±1.66	88.88±1.86	88.48±1.99	88.46±1.98
	Agrawal	77.27±4.79	76.56±5.24	78.05±5.25	78.18±5.37	78.60±5.25
Gradual-500K	RBF	61.47±4.12	60.11±3.85	61.02±4.45	60.80±4.47	61.82±4.21
	Asset Negotiation	88.16±1.90	87.69±1.71	88.78±1.84	88.35±2.00	88.35±2.00
	Adult Stream	74.47±1.50	72.16±0.22	75.15±0.73	75.40±0.59	75.19±0.66
	Airlines	67.71±1.60	64.72±1.07	66.75±1.13	66.91±1.23	68.23±1.05
	Avila	31.94±0.49	31.25±0.78	32.32±0.49	33.74±0.79	31.39±0.64
Real World	Codrna	73.39±3.38	68.94±4.17	71.08±4.42	71.34±4.62	72.47±4.19
	Electricity	82.48±2.54	87.40±0.94	85.43±0.87	83.90±0.72	85.56±0.61
	Cover Type	86.62±3.42	89.28±3.04	86.88±2.55	84.50±2.30	86.60±2.69
	Poker hand	65.84±5.22	79.81±0.78	77.84±0.77	77.32±0.74	77.09±0.91
	Online Shoppers	83.67±2.10	83.14±2.18	83.47±2.69	82.33±1.78	83.02±1.22
RANK		5.86	7.72	4.25	4.30	3.19
-- table continued next						

From the tested results on both real and artificial datasets (referring to Table 5 and Table 6), SCMER produced higher accuracies in most of the scenarios. The overall accuracy comparison shows that SCMER achieved a ranking among the top three algorithms, showing outstanding performance with respective to classification.

When using the same classifier, SCMER achieved optimal performance of classification accuracy in many scenarios. While using the Naive Bayes classifier, the ranking of SCMER is 2.70 which is shown in the table 5. Along with SCMER the next best performing algorithms are HDDM_A_Test and RDDM which are ranked 3.19 and 3.44 respectively. Additionally, the lowest ranking algorithm is ECDD which is 7.72. Similarly, in the case of Hoeffding Tree classifier, SCMER has a ranking of 3.06, followed by FHDDM and HDDM_A_Test with a ranking of 3.23. While VFDD showed the lowest rank of 7.80. From the accuracy rankings, SCMER, RDDM, FHDDM and HDDM_A_Test were the top 50% rankings among all the datasets.

In case of artificial datasets, the drift locations are known and the performance of the classifier depends on the detections of these drifts so to reset and adapt themselves for new data. This can be seen in the classification results provided in table 5 and 6. For the real-world datasets, the number of drift occurrence is unknown which causes problem for the analysis of the drift detection algorithms. Looking at the accuracy results, in most real-world datasets, ECDD produced the the highest accuracies, indicating it is a good-performing algorithm, but it is the worst-performing algorithm in artificial datasets. This indicates that ECDD made more resets on the base classifier by detecting a higher number of false drifts which is not ideal in real scenarios. In case of the SCMER algorithm, it produced accuracies near to the ECDD algorithm, which indicate that proposed method made the least detections and coped up to ECDD. For the better analysis of the algorithm, only classification performance cannot be used; hence, drift detection analysis on artificial datasets is done for better understanding of the algorithms which is provided in next section.

The accuracy analysis indicates that the SCMER algorithm showed determined ability to detect drifts, which improved accuracies of the base classifiers in different scenarios of datasets.

Table 5: Average Accuracies in Percentage (%) Using Naive Bayes, with 95% Confidence Intervals in the Artificial Datasets. (-- Table Continued)

TYPE-SIZE	DATASET	HDDM_W_Test	RDDM	VFDD	SCMER
Abrupt-50K	Agrawal	78.35±5.20	78.16±4.81	78.57±5.14	78.92±5.03
	RBF	60.14±4.57	61.88±4.17	59.99±4.72	61.97±4.18
	Asset Negotiation	84.39±1.24	88.67±2.03	87.90±1.56	88.80±2.04
Abrupt-100K	Agrawal	78.54±5.21	78.35±4.99	78.41±5.22	78.86±5.09
	RBF	60.05±4.65	61.73±4.05	59.81±4.49	61.80±4.02
	Asset Negotiation	85.48±1.15	88.76±2.06	88.14±1.64	88.86±2.08
Abrupt-300K	Agrawal	77.71±5.45	77.92±5.32	77.08±5.31	78.04±5.48
	RBF	59.22±4.60	61.15±4.31	59.33±4.91	61.29±4.26
	Asset Negotiation	87.50±1.77	88.45±2.01	87.90±1.51	88.46±2.01
Abrupt-500K	Agrawal	78.49±5.31	78.60±5.25	77.56±5.21	78.59±5.28
	RBF	60.17±4.43	61.23±4.14	59.90±4.50	61.81±4.16
	Asset Negotiation	88.12±1.84	88.57±1.93	87.93±1.43	88.37±1.99
Gradual-50K	Agrawal	77.59±4.78	78.11±4.79	78.49±5.10	78.71±4.96
	RBF	60.59±4.48	61.22±3.93	59.22±4.70	61.38±4.01
	Asset Negotiation	84.69±1.89	88.63±2.07	87.65±1.69	88.73±2.07
Gradual-100K	Agrawal	78.03±5.29	78.34±4.98	77.65±5.08	78.70±5.05
	RBF	61.02±3.78	61.12±3.79	59.30±4.86	61.25±3.84
	Asset Negotiation	84.34±1.89	88.73±2.08	88.20±1.58	88.82±2.08
Gradual-300K	Agrawal	77.32±5.52	78.11±5.46	76.92±5.32	78.02±5.46
	RBF	59.57±4.60	60.93±4.23	59.30±4.70	61.11±4.22
	Asset Negotiation	88.17±1.72	88.75±1.99	87.83±1.44	88.48±1.99
Gradual-500K	Agrawal	78.27±5.36	78.61±5.26	77.43±5.14	78.55±5.26
	RBF	60.53±4.40	61.19±4.12	59.74±4.42	61.72±4.12
	Asset Negotiation	88.34±1.95	88.70±1.98	87.68±1.61	88.35±2.00
Real World	Adult Stream	74.75±0.76	75.28±0.53	72.78±0.27	75.40±0.57
	Airlines	66.98±1.26	68.58±1.15	65.66±1.21	67.86±1.19
	Avila	33.05±0.43	31.48±0.61	33.88±0.31	32.91±0.66
	Codrna	69.91±4.90	72.88±3.66	67.50±4.14	72.68±4.19
	Electricity	81.57±0.95	85.06±0.73	84.78±0.75	83.05±0.81
	Cover Type	82.43±2.58	86.18±2.46	86.61±2.45	83.72±2.19
	Poker hand	74.20±0.65	77.27±0.78	77.62±0.69	76.35±0.80
	Online Shoppers	81.89±1.83	84.12±1.99	81.33±3.38	82.74±1.46
RANK		6.58	3.44	6.97	2.70

Table 6: Average Accuracies in Percentage (%) Using Hoeffding Tree, with 95% Confidence Intervals in the Artificial Datasets

TYPE-SIZE	DATASET	DDM	ECDD	STEPD	FHDDM	HDDM_A_Test
Abrupt-50K	Agrawal	82.83±3.90	77.18±5.53	85.15±4.58	85.44±4.74	85.02±4.53
	RBF	64.12±5.83	58.87±3.75	59.98±5.75	62.65±5.78	64.21±5.81
	Asset Negotiation	92.92±1.33	89.15±1.33	92.23±1.18	93.01±1.36	92.99±1.36
Abrupt-100K	Agrawal	87.71±4.36	77.36±5.49	84.91±3.86	87.30±4.46	87.05±4.32
	RBF	65.13±6.37	59.19±3.67	60.34±5.10	63.77±6.47	65.18±6.39
	Asset Negotiation	93.67±1.08	89.81±1.18	92.95±0.98	93.79±1.12	93.82±1.12
Abrupt-300K	Agrawal	89.90±3.53	76.44±5.60	86.65±3.92	89.57±3.91	89.41±3.82
	RBF	68.85±8.10	59.52±3.96	63.34±7.23	65.19±8.22	66.10±7.90
	Asset Negotiation	94.11±0.70	89.16±1.01	93.29±0.62	94.20±0.72	94.18±0.72
Abrupt-500K	Agrawal	90.96±3.20	77.21±5.45	86.41±3.29	90.67±3.52	90.53±3.45
	RBF	73.07±8.04	60.06±3.84	63.20±5.96	66.63±8.18	67.34±7.63
	Asset Negotiation	94.25±0.55	89.04±1.03	93.32±0.48	94.31±0.57	94.31±0.57
Gradual-50K	Agrawal	85.10±4.59	77.17±5.53	84.46±4.27	85.31±4.67	84.97±4.50
	RBF	61.16±3.93	58.94±3.76	59.35±5.54	61.82±5.51	61.49±4.04
	Asset Negotiation	92.91±1.36	89.05±1.34	92.21±1.20	92.96±1.37	92.97±1.38
Gradual-100K	Agrawal	87.37±4.20	77.34±5.47	84.94±3.89	87.24±4.43	86.98±4.30
	RBF	61.06±3.77	59.14±3.64	59.85±4.95	63.33±6.34	62.09±4.62
	Asset Negotiation	93.66±1.10	89.39±1.29	92.84±0.96	93.76±1.13	93.76±1.13
Gradual-300K	Agrawal	89.91±3.53	76.44±5.61	86.64±3.91	89.42±3.82	89.56±3.90
	RBF	66.66±6.92	58.87±3.79	63.28±7.21	65.07±8.19	64.65±7.20
	Asset Negotiation	94.10±0.69	89.28±0.91	93.29±0.61	94.19±0.71	94.17±0.71
Gradual-500K	Agrawal	90.96±3.20	77.20±5.45	86.51±3.36	90.54±3.46	90.52±3.45
	RBF	70.92±6.98	60.05±3.83	63.15±5.94	66.52±8.15	66.74±7.54
	Asset Negotiation	94.22±0.55	89.15±1.00	93.25±0.48	94.31±0.57	94.31±0.57
Real World	Adult Stream	75.50±1.63	71.93±0.24	74.82±0.44	76.01±0.38	77.09±0.34
	Airlines	66.41±1.14	65.24±1.29	66.76±1.32	66.68±1.31	66.50±1.33
	Avila	44.07±2.46	44.03±2.45	42.55±1.18	43.83±2.43	44.03±2.45
	Codrna	76.43±2.59	68.04±3.84	69.60±4.00	71.15±4.45	71.47±3.84
	Electricity	86.70±0.96	87.27±0.99	86.74±1.19	86.40±1.59	86.91±1.02
	Cover Type	87.24±2.94	88.95±3.05	86.41±2.46	84.47±2.33	86.53±2.64
	Poker hand	73.69±1.53	79.30±0.78	77.87±0.85	77.40±0.81	77.10±0.83
	Online Shoppers	91.06±1.77	88.70±2.46	89.49±1.84	91.09±1.65	91.02±1.65
RANK		3.36	7.67	6.38	3.23	3.23

-- table continued next

Table 6: Average Accuracies in Percentage (%) Using Hoeffding Tree, with 95% Confidence Intervals in the Artificial Datasets. (-- Table Continued)

TYPE-SIZE	DATASET	HDDM_W_Test	RDDM	VFDD	SCMER
Abrupt-50K	Agrawal	85.49±4.76	85.17±4.64	84.85±4.80	85.35±4.70
	RBF	61.89±5.85	64.34±5.90	60.29±4.79	64.39±5.88
	Asset Negotiation	92.97±1.34	92.93±1.34	88.34±1.17	92.99±1.36
Abrupt-100K	Agrawal	85.10±3.92	87.10±4.39	86.80±4.48	86.92±4.26
	RBF	63.49±6.07	65.12±6.32	59.92±5.18	64.16±5.77
	Asset Negotiation	93.59±1.06	93.75±1.11	88.84±1.06	93.79±1.12
Abrupt-300K	Agrawal	88.19±4.47	89.34±3.79	87.28±3.64	89.55±3.90
	RBF	60.74±5.22	69.05±8.19	59.47±4.83	66.10±7.79
	Asset Negotiation	93.78±0.67	94.02±0.70	88.58±1.11	94.20±0.72
Abrupt-500K	Agrawal	86.58±4.30	90.45±3.42	87.30±3.21	90.52±3.45
	RBF	62.03±5.29	72.58±7.69	59.97±4.55	67.64±7.84
	Asset Negotiation	93.85±0.49	94.14±0.55	88.80±1.08	94.31±0.57
Gradual-50K	Agrawal	84.45±4.25	84.38±4.23	84.22±4.49	85.32±4.68
	RBF	60.81±4.19	61.20±3.94	58.91±4.71	62.02±4.51
	Asset Negotiation	92.57±1.31	92.92±1.37	88.32±1.30	92.98±1.38
Gradual-100K	Agrawal	85.76±4.36	86.87±4.26	86.81±4.49	86.91±4.26
	RBF	59.71±4.71	61.11±3.80	59.57±4.71	62.20±4.60
	Asset Negotiation	93.34±1.07	93.73±1.12	89.23±1.17	93.75±1.12
Gradual-300K	Agrawal	88.00±4.34	89.34±3.78	87.28±3.64	89.55±3.89
	RBF	59.72±4.80	66.13±6.51	59.55±4.78	65.81±7.78
	Asset Negotiation	94.07±0.68	94.03±0.69	88.56±1.03	94.19±0.71
Gradual-500K	Agrawal	87.72±3.38	90.43±3.40	84.49±2.95	90.51±3.44
	RBF	62.08±5.70	70.46±6.71	59.17±4.40	67.28±7.80
	Asset Negotiation	93.86±0.51	94.11±0.56	88.74±1.02	94.31±0.57
Real World	Adult Stream	74.04±0.67	76.83±0.42	73.27±0.38	76.94±0.30
	Airlines	66.35±1.25	67.04±1.17	66.41±1.34	67.02±1.15
	Avila	40.84±0.53	44.03±2.45	42.33±1.61	44.03±2.45
	Codrna	69.51±4.96	71.88±3.48	67.42±4.43	71.76±3.91
	Electricity	85.39±2.08	86.91±1.38	85.91±1.20	86.11±1.74
	Cover Type	83.63±2.41	85.83±2.37	86.19±2.53	83.89±2.30
	Poker hand	73.45±0.83	77.40±0.87	77.54±0.77	76.74±0.92
	Online Shoppers	91.09±1.65	91.05±1.85	89.13±2.36	91.09±1.65
RANK		6.38	3.89	7.80	3.06

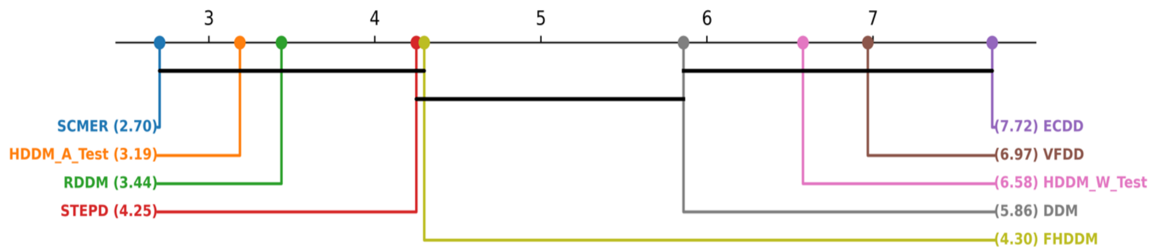


Fig. 2: Comparison Results Using the Bonferroni Post-Hoc Test on All Tested Datasets for Naive Bayes Classifier.

Fig. 3: Comparison Results Using the Bonferroni Post-Hoc Test on All Tested Datasets for Hoeffding Tree Classifier.

For a statistical comparison of the accuracy results, we applied the Friedman test, a non-parametric statistical method [35]. The null hypothesis for this test states that all compared methods are statistically equivalent. If this hypothesis is rejected, a post-hoc test is required to identify which specific methods differ significantly.

We used the Bonferroni-Dunn post-hoc test [35] to compare the SCMER method with other state-of-the-art methods. The results of this test are presented graphically in Figures 2 and 3. The critical difference (CD) is represented by a horizontal bar; any methods connected by this bar are not statistically different from each other.

As illustrated in Figures 2 and 3, which visually represents the evaluation of the concept drift detection methods based on the data in Table 5 and Table 6 respectively, SCMER achieved the highest rank in this set of experiments. The next two best-performing methods, HDDM_A-Test and RDDM in case of Naive Bayes and for Hoeffding Tree, HDDM_A-Test and FHDDM, are not statistically different from each other.

The accuracy results for both artificial and real-world datasets for Naive Bayes Classifier and Hoeffding Tree Classifier are represented graphically in figures 4 through 7 for Naive Bayes Classifier and figures 8 through 11 for Hoeffding Tree Classifier. In Figures 4, 5, 6, 8, 9 and 10, the horizontal red lines indicate the actual positions where concept drifts occur within the datasets.

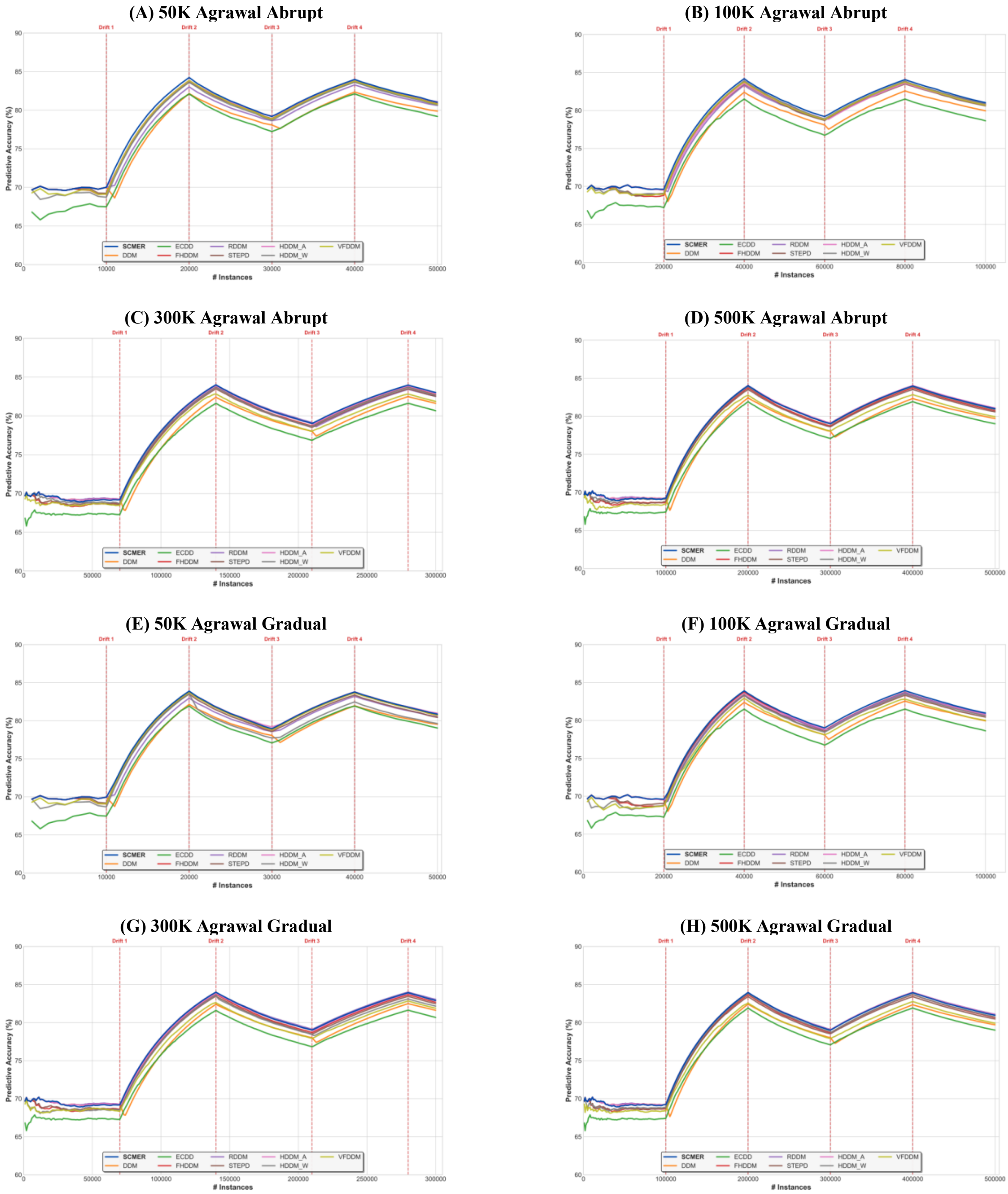
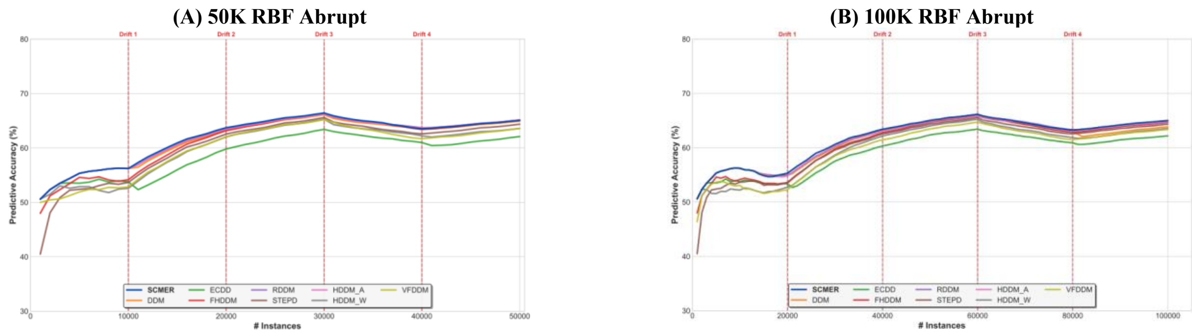
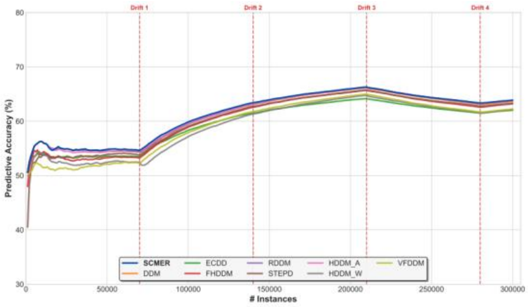


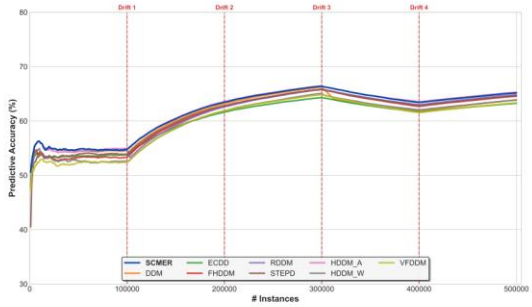
Fig. 4: Accuracy Results in Case of Agrawal Dataset for Naive Bayes Classifier.



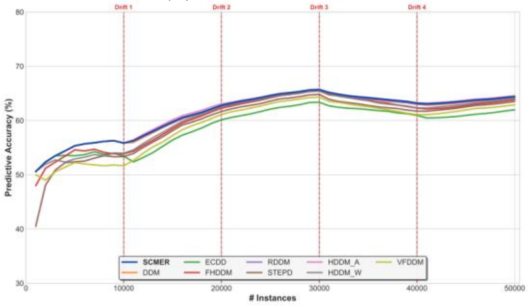
(C) 300K RBF Abrupt

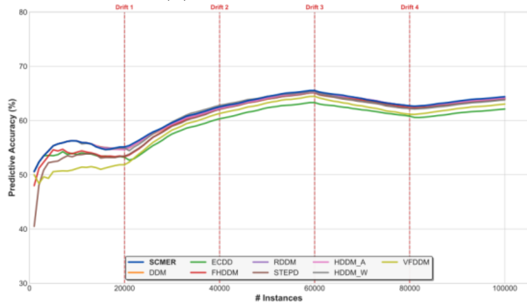
(D) 500K RBF Abrupt



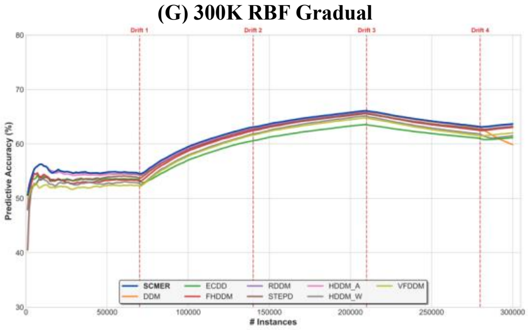


(E) 50K RBF Gradual **(F) 100K RBF Gradual**





(G) 300K RBF Gradual **(H) 500K RBF Gradual**



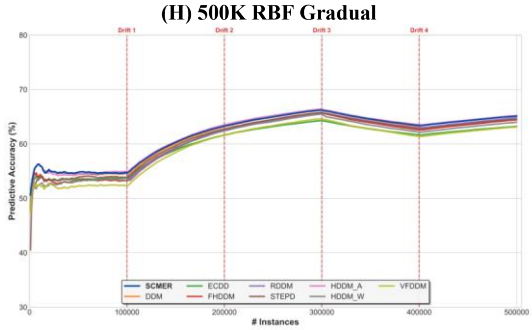
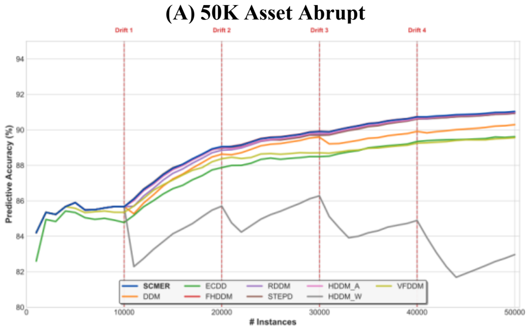
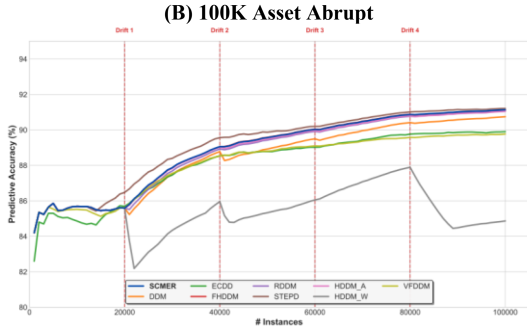


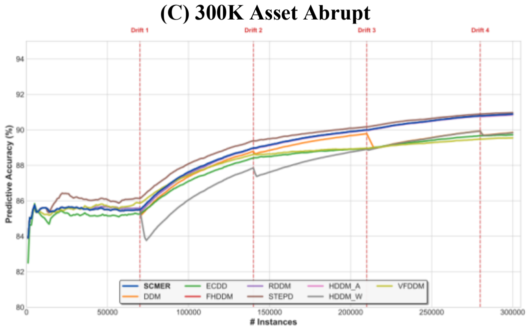
Fig. 5: Accuracy Results in Case of RBF Dataset for Naive Bayes Classifier.

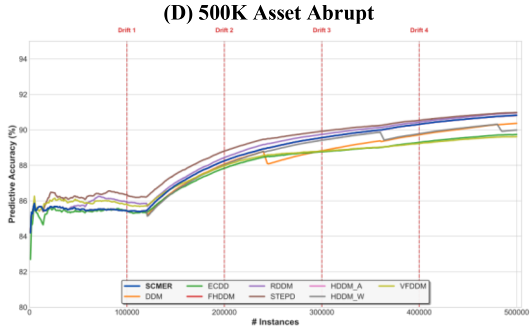
(A) 50K Asset Abrupt **(B) 100K Asset Abrupt**



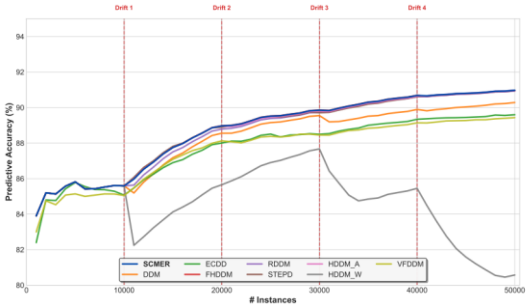


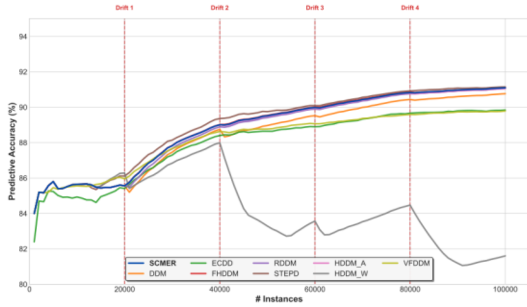
(C) 300K Asset Abrupt **(D) 500K Asset Abrupt**



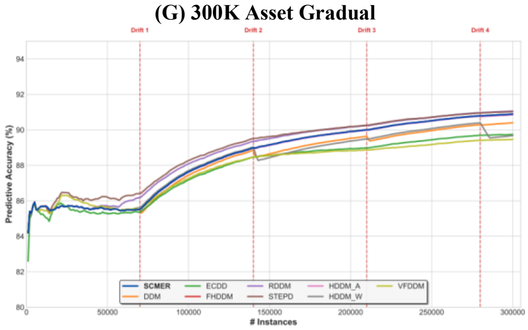


(E) 50K Asset Gradual **(F) 100K Asset Gradual**





(G) 300K Asset Gradual **(H) 500K Asset Gradual**



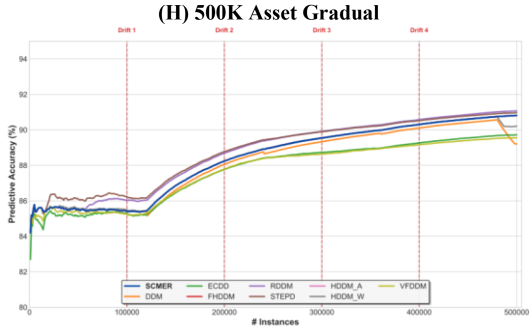
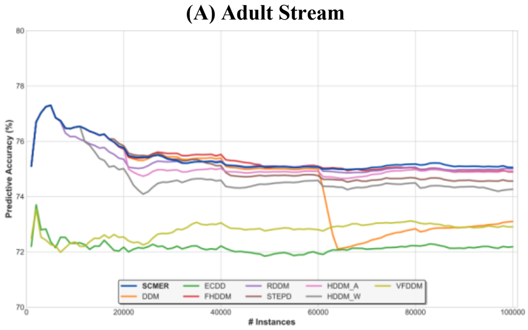
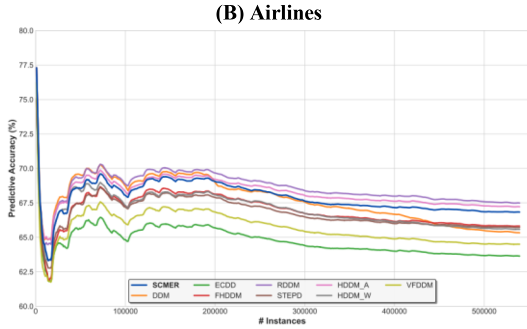


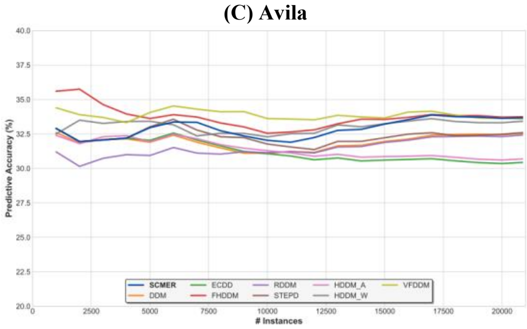
Fig. 6: Accuracy Results in Case of Asset Negotiation Dataset for Naive Bayes Classifier.

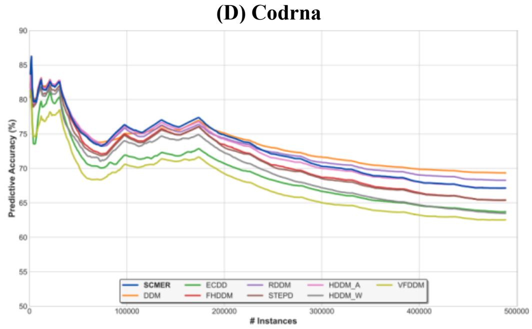
(A) Adult Stream **(B) Airlines**



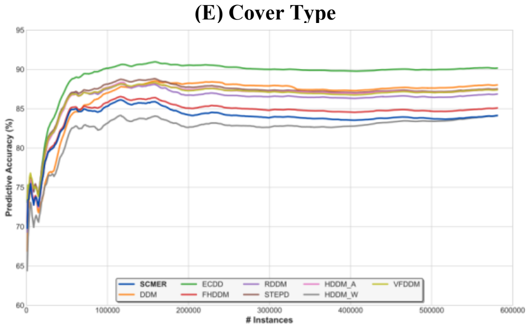


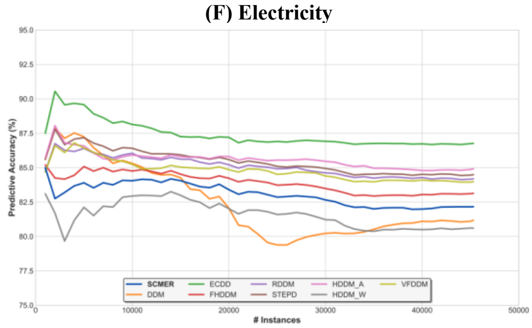
(C) Avila **(D) Codrna**





(E) Cover Type **(F) Electricity**





(G) Online Shoppers **(H) Poker Hand**

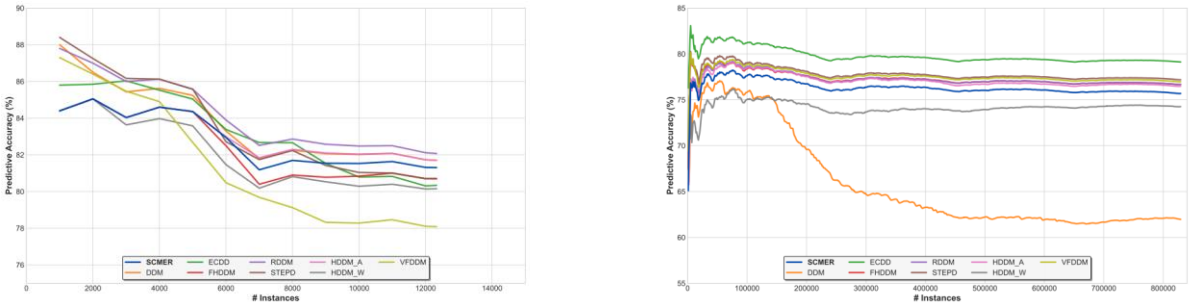


Fig. 7: Accuracy Results in Case of Real-World Dataset for Naive Bayes Classifier.

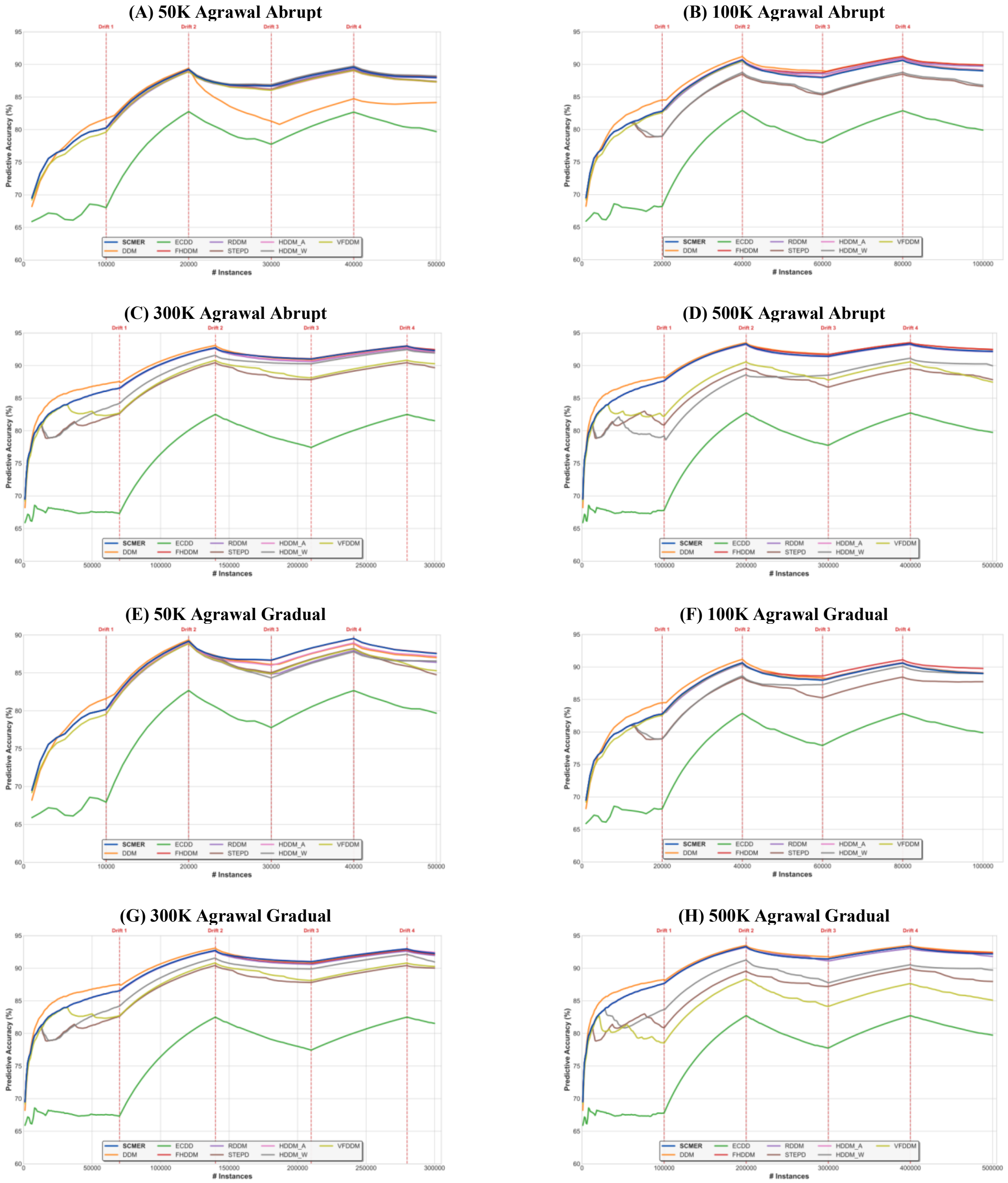
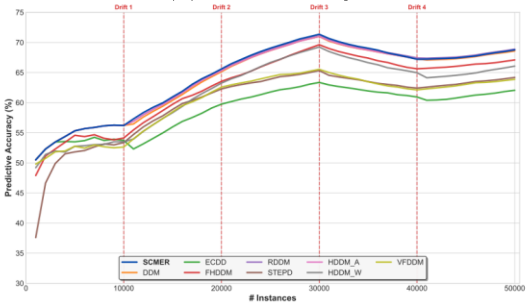
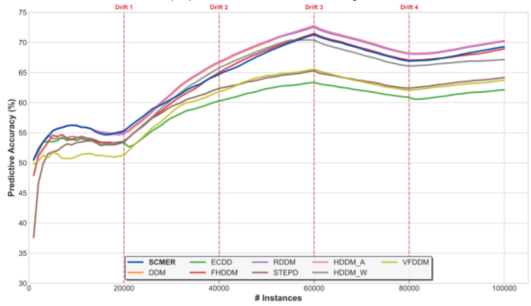


Fig. 8: Accuracy Results in Case of Agrawal Dataset for Hoeffding Tree Classifier.

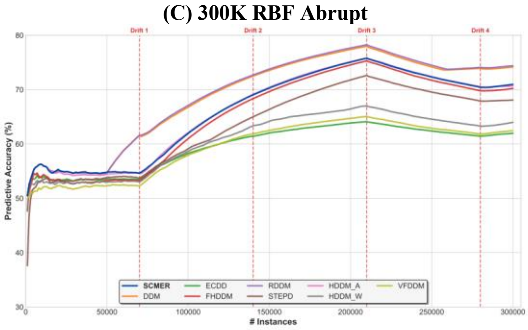
(A) 50K RBF Abrupt



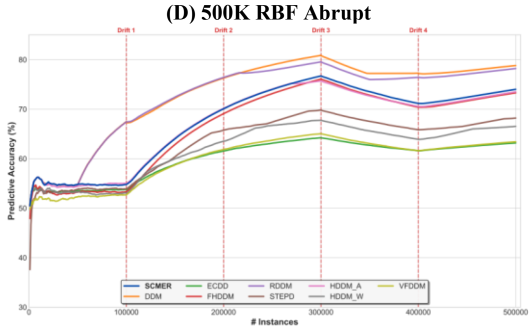
(B) 100K RBF Abrupt



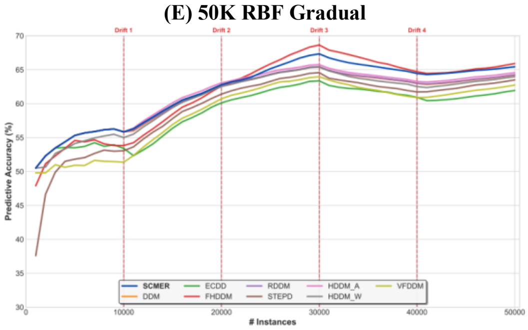
(C) 300K RBF Abrupt



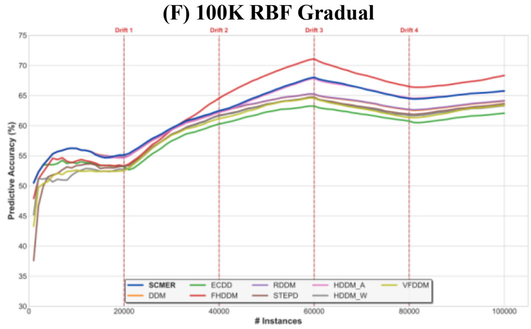
(D) 500K RBF Abrupt



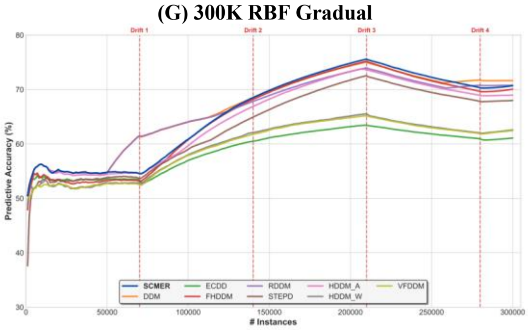
(E) 50K RBF Gradual



(F) 100K RBF Gradual



(G) 300K RBF Gradual



(H) 500K RBF Gradual

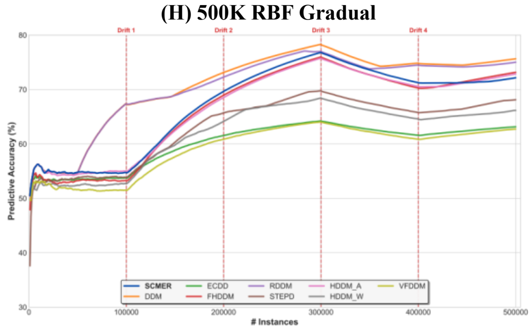
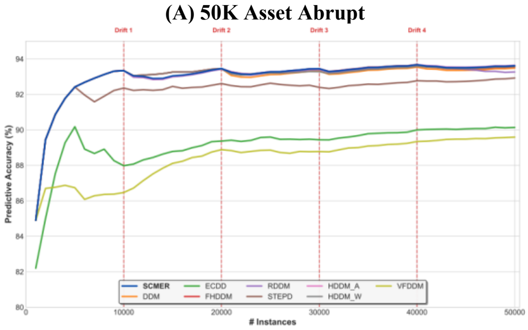
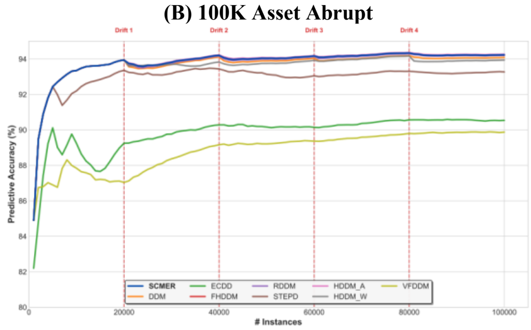


Fig. 9: Accuracy Results in Case of RBF Dataset for Hoeffding Tree Classifier.

(A) 50K Asset Abrupt

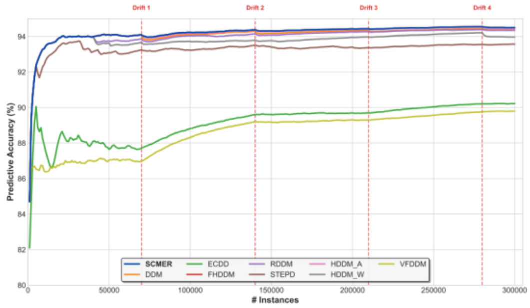


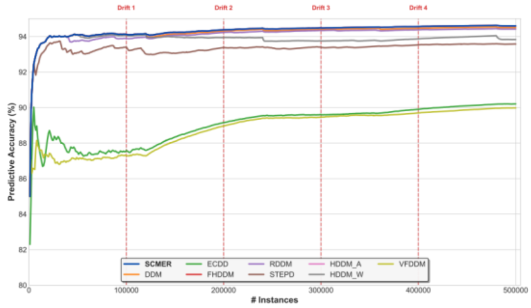
(B) 100K Asset Abrupt



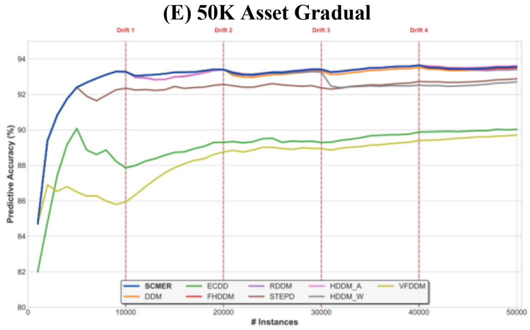
(C) 300K Asset Abrupt

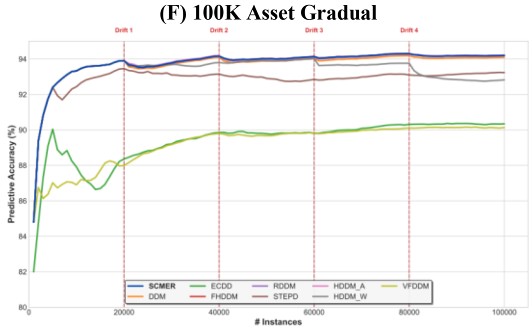
(D) 500K Asset Abrupt



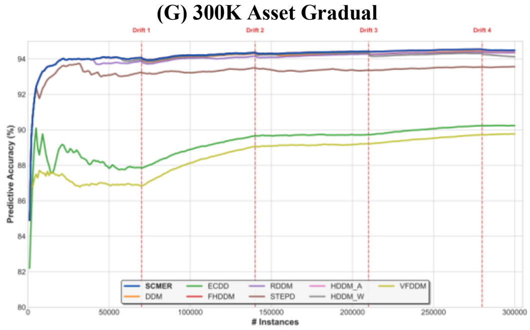


(E) 50K Asset Gradual **(F) 100K Asset Gradual**





(G) 300K Asset Gradual **(H) 500K Asset Gradual**



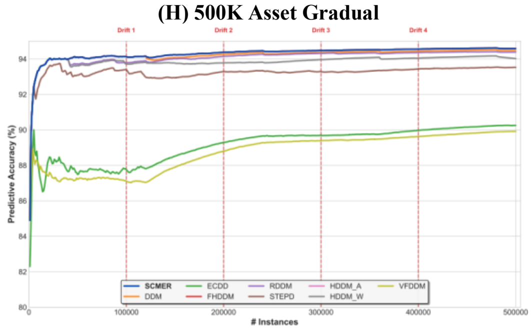
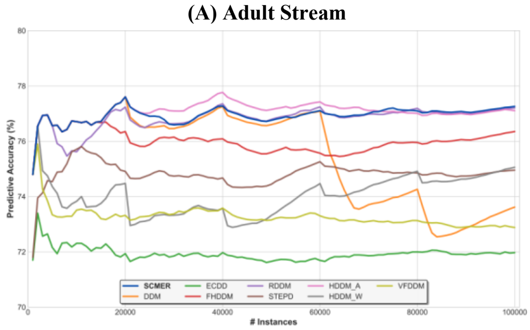
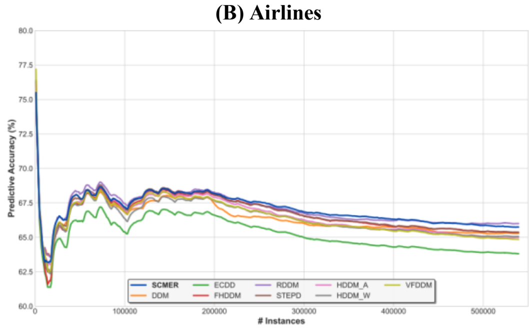


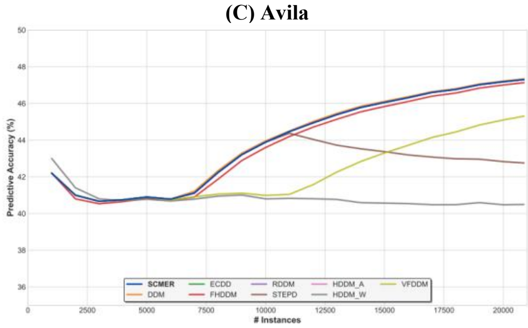
Fig. 10: Accuracy Results in Case of Asset Negotiation Dataset for Hoeffding Tree Classifier.

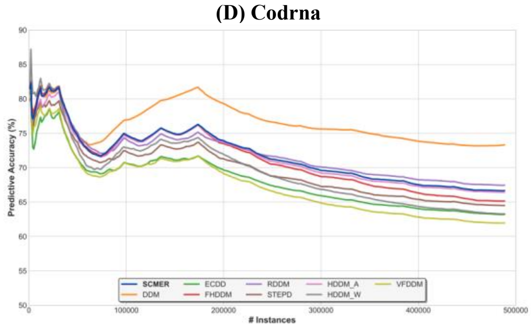
(A) Adult Stream **(B) Airlines**



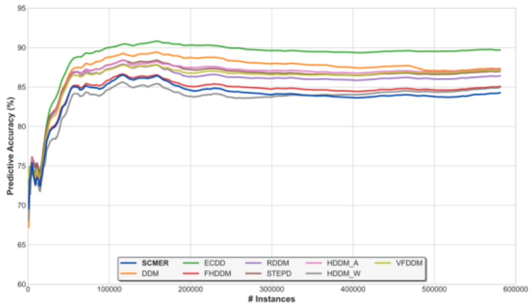


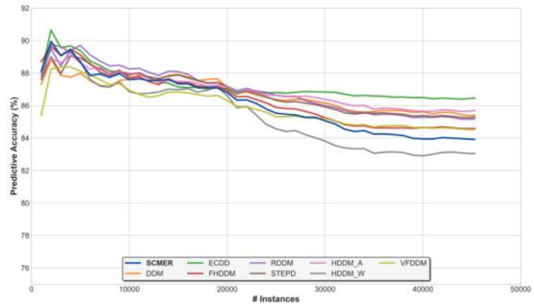
(C) Avila **(D) Codrna**



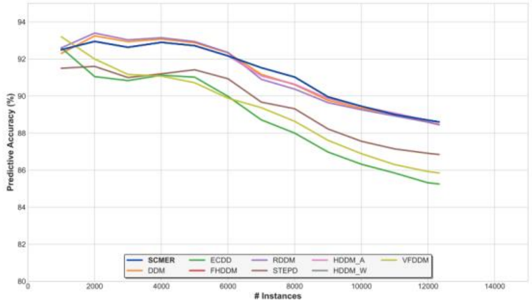


(E) Cover Type **(F) Electricity**





(G) Online Shoppers **(H) Poker Hand**



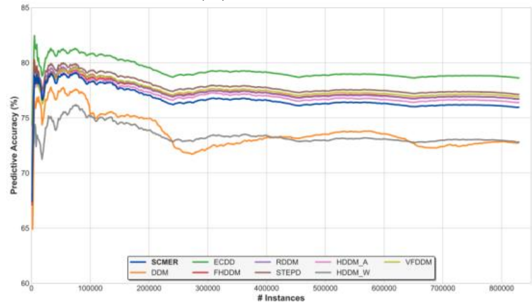


Fig. 11: Accuracy Results in Case of Real-World Dataset for Hoeffding Tree Classifier.

5.2. Drift identification analysis

We evaluated the performance of the concept drift detectors by analysing their ability to identify drifts. This was measured by comparing the detected drift positions and the number of drifts identified by each method.

Table 7 and Table 8 presents the mean distance (μDist) to the correct drift location, along with True Positive (TP), False Positive (FP), and False Negative (FN) rates for each method on datasets with abrupt drifts, as the true drift positions are known for these.} In contrast, for datasets with gradual concept drifts, it is difficult to pinpoint a single drift location, which complicates the computation of these metrics. In this context, false positives are incorrectly identified drifts, while false negatives are actual drifts that the methods failed to detect. In both cases, a lower number is desirable.

Table 7: Concept Drift Identifications in the Abrupt Datasets Using Naive Bayes as Base Classifier.

Dataset	Algorithm	μDist	FN	FP	Precision	Recall	F1	MCC
	DDM	36	2.00	3.00	0.40	0.50	0.44	0.45
	ECDD	1.5	2.00	85.00	0.02	0.50	0.04	0.11
	HDDM_A	52.5	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	20.25	0.00	12.00	0.25	1.00	0.40	0.50
50K Agrawal	STEPD	13.75	0.00	10.00	0.29	1.00	0.44	0.53
	FHDDM	18.25	0.00	9.00	0.31	1.00	0.47	0.55
	RDDM	79.33	1.00	1.00	0.75	0.75	0.75	0.75
	VFDD	10.75	0.00	14.00	0.22	1.00	0.36	0.47
	SCMER	30	0.00	0.00	1.00	1.00	1.00	1.00
	DDM	79.5	1.00	2.00	0.60	0.75	0.67	0.67
	ECDD	8.25	0.00	69.00	0.05	1.00	0.10	0.23
	HDDM_A	54.25	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	32.25	0.00	0.00	1.00	1.00	1.00	1.00
50K Asset Negotiation	STEPD	13.75	0.00	9.00	0.31	1.00	0.47	0.55
	FHDDM	33.5	0.00	0.00	1.00	1.00	1.00	1.00
	RDDM	77	0.00	1.00	0.80	1.00	0.89	0.89
	VFDD	23.75	3.00	57.00	0.05	0.75	0.09	0.19
	SCMER	45.25	0.00	0.00	1.00	1.00	1.00	1.00
	DDM	176	2.00	1.00	0.67	0.50	0.57	0.58
	ECDD	5	3.00	39.00	0.03	0.25	0.05	0.08
	HDDM_A	89.67	1.00	0.00	1.00	0.75	0.86	0.87
	HDDM_W	12.34	1.00	30.00	0.09	0.75	0.16	0.26
50K RBF	STEPD	14.67	1.00	34.00	0.08	0.75	0.15	0.25
	FHDDM	11.34	1.00	9.00	0.25	0.75	0.38	0.43
	RDDM	109.67	1.00	0.00	1.00	0.75	0.86	0.87
	VFDD	16.67	1.00	37.00	0.08	0.75	0.14	0.24
	SCMER	28.66	1.00	0.00	1.00	0.75	0.86	0.87
	DDM	48.5	2.00	3.00	0.40	0.50	0.44	0.45
	ECDD	1.5	2.00	222.00	0.01	0.50	0.02	0.07
	HDDM_A	175.5	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	16.5	2.00	21.00	0.09	0.50	0.15	0.21
100K Agrawal	STEPD	7.75	0.00	24.00	0.14	1.00	0.25	0.38
	FHDDM	14	0.00	18.00	0.18	1.00	0.31	0.43
	RDDM	64.66	1.00	3.00	0.50	0.75	0.60	0.61
	VFDD	5.75	0.00	30.00	0.12	1.00	0.21	0.34
	SCMER	31.5	0.00	0.00	1.00	1.00	1.00	1.00
100K Asset Negotiation	DDM	200	1.00	3.00	0.50	0.75	0.60	0.61

	ECDD	11.25	0.00	134.00	0.03	1.00	0.06	0.17
	HDDM_A	102	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	31.75	0.00	0.00	1.00	1.00	1.00	1.00
	STEPD	17	0.00	22.00	0.15	1.00	0.27	0.39
	FHDDM	34.5	0.00	0.00	1.00	1.00	1.00	1.00
	RDDM	109.75	0.00	0.00	1.00	1.00	1.00	1.00
	VFDD	26.25	0.00	126.00	0.03	1.00	0.06	0.18
	SCMER	53.5	0.00	0.00	1.00	1.00	1.00	1.00
	DDM	215.5	2.00	4.00	0.33	0.50	0.40	0.41
	ECDD	279	3.00	109.00	0.01	0.25	0.02	0.05
	HDDM_A	22.34	1.00	5.00	0.38	0.75	0.50	0.53
	HDDM_W	64.5	0.00	52.00	0.07	1.00	0.13	0.27
100K RBF	STEPD	9	1.00	55.00	0.05	0.75	0.10	0.20
	FHDDM	9.34	1.00	32.00	0.09	0.75	0.15	0.25
	RDDM	93.34	1.00	4.00	0.43	0.75	0.55	0.57
	VFDD	8.34	1.00	70.00	0.04	0.75	0.08	0.18
	SCMER	30	0.00	2.00	0.67	1.00	0.80	0.82
-- table continued on next								

Table 7: Concept Drift Identifications in the Abrupt Datasets Using Naive Bayes as Base Classifier. (-- Table Continued from Previous)

Dataset	Algorithm	µDist	FN	FP	Precision	Recall	F1	MCC
	DDM	97.00	2.00	24.00	0.08	0.50	0.13	0.20
	ECDD	2.75	0.00	756.00	0.01	1.00	0.01	0.07
	HDDM_A	243.25	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	12.75	0.00	55.00	0.07	1.00	0.13	0.26
300K Agrawal	STEPD	8.75	0.00	76.00	0.05	1.00	0.10	0.22
	FHDDM	16.25	0.00	48.00	0.08	1.00	0.14	0.28
	RDDM	39.33	1.00	17.00	0.15	0.75	0.25	0.34
	VFDD	8.25	0.00	80.00	0.05	1.00	0.09	0.22
	SCMER	90.50	0.00	12.00	0.25	1.00	0.40	0.50
	DDM	891.33	2.00	6.00	0.25	0.50	0.33	0.35
	ECDD	267.25	0.00	402.00	0.01	1.00	0.02	0.10
	HDDM_A	114.75	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	26.00	0.00	0.00	1.00	1.00	1.00	1.00
300KAsset Negotiation	STEPD	13.75	0.00	62.00	0.06	1.00	0.11	0.25
	FHDDM	28.25	0.00	0.00	1.00	1.00	1.00	1.00
	RDDM	155.75	0.00	11.00	0.27	1.00	0.42	0.52
	VFDD	27.75	0.00	330.00	0.01	1.00	0.02	0.11
	SCMER	47.75	0.00	0.00	1.00	1.00	1.00	1.00
	DDM	411.00	1.00	2.00	0.60	0.75	0.67	0.67
	ECDD	68.75	0.00	323.00	0.01	1.00	0.02	0.11
	HDDM_A	259.00	1.00	10.00	0.23	0.75	0.35	0.42
	HDDM_W	14.34	1.00	152.00	0.02	0.75	0.04	0.12
300K RBF	STEPD	10.00	1.00	110.00	0.03	0.75	0.05	0.14
	FHDDM	12.34	1.00	103.00	0.03	0.75	0.05	0.15
	RDDM	178.00	1.00	17.00	0.15	0.75	0.25	0.34
	VFDD	17.00	1.00	171.00	0.02	0.75	0.03	0.11
	SCMER	20.00	1.00	11.00	0.21	0.75	0.33	0.40
	DDM	105.00	2.00	14.00	0.13	0.50	0.20	0.25
	ECDD	552.25	0.00	1146.00	0.00	1.00	0.01	0.06
	HDDM_A	19.25	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	9.75	0.00	109.00	0.04	1.00	0.07	0.19
500K Agrawal	STEPD	9.50	0.00	142.00	0.03	1.00	0.05	0.17
	FHDDM	18.00	0.00	96.00	0.04	1.00	0.08	0.20
	RDDM	142.00	0.00	29.00	0.12	1.00	0.22	0.35
	VFDD	7.00	0.00	129.00	0.03	1.00	0.06	0.17
	SCMER	139.25	0.00	12.00	0.25	1.00	0.40	0.50
	DDM	490.66	1.00	5.00	0.38	0.75	0.50	0.53
	ECDD	8.75	0.00	672.00	0.01	1.00	0.01	0.08
	HDDM_A	43.75	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	31.25	0.00	0.00	1.00	1.00	1.00	1.00
500K Asset Negotiation	STEPD	14.00	0.00	110.00	0.04	1.00	0.07	0.19
	FHDDM	34.50	0.00	0.00	1.00	1.00	1.00	1.00
	RDDM	208.25	0.00	19.00	0.17	1.00	0.30	0.42
	VFDD	511.00	0.00	521.00	0.01	1.00	0.02	0.09
	SCMER	59.25	0.00	0.00	1.00	1.00	1.00	1.00
	DDM	676.50	2.00	2.00	0.50	0.50	0.50	0.50
	ECDD	102.50	0.00	574.00	0.01	1.00	0.01	0.08
	HDDM_A	342.00	1.00	14.00	0.22	0.80	0.35	0.42
	HDDM_W	152.25	0.00	248.00	0.02	1.00	0.03	0.13
500K RBF	STEPD	11.34	1.00	190.00	0.02	0.75	0.03	0.11
	FHDDM	10.34	1.00	159.00	0.02	0.75	0.04	0.12
	RDDM	216.33	1.00	26.00	0.10	0.75	0.18	0.28
	VFDD	121.50	2.00	269.00	0.01	0.50	0.01	0.06
	SCMER	25.00	1.00	17.00	0.15	0.75	0.25	0.34

We classified drift identifications into different categories, considering a detection as a true positive only if it occurred within a 2% window of the concept size following the actual drift point.

As shown in Table 7 and Table 8, the ECDD and STEPD approaches yielded the best results in terms of mean distance to the true drift. Regarding false negatives, ECDD and STEPD also performed best for most datasets. However, these two methods produced a higher number of false positives, which negatively impacts accuracy.

Table 8: Concept Drift Identifications in the Abrupt Datasets Using Hoeffding Tree as Base Classifier

Dataset	Algorithm	µdist	FN	FP	Precision	Recall	F1	MCC
50K Agrawal	DDM	93.34	1.00	10.00	0.23	0.75	0.35	0.42
	ECDD	3.75	0.00	128.00	0.03	1.00	0.06	0.17
	HDDM_A	48.25	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	17.75	0.00	0.00	1.00	1.00	1.00	1.00
	STEPD	9.75	0.00	8.00	0.33	1.00	0.50	0.58
	FHDDM	18.25	0.00	0.00	1.00	1.00	1.00	1.00
	RDDM	82.75	0.00	0.00	1.00	1.00	1.00	1.00
	VFDDM	11.75	0.00	3.00	0.57	1.00	0.73	0.76
	SCMER	34.50	0.00	0.00	1.00	1.00	1.00	1.00
50K Asset Negotiation	DDM	128.00	1.00	1.00	0.75	0.75	0.75	0.75
	ECDD	8.25	0.00	61.00	0.06	1.00	0.12	0.25
	HDDM_A	38.75	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	32.50	0.00	0.00	1.00	1.00	1.00	1.00
	STEPD	10.75	0.00	6.00	0.40	1.00	0.57	0.63
	FHDDM	58.67	1.00	0.00	1.00	0.75	0.86	0.87
	RDDM	68.25	0.00	3.00	0.57	1.00	0.73	0.76
	VFDDM	32.00	1.00	62.00	0.05	0.75	0.09	0.19
	SCMER	45.50	0.00	0.00	1.00	1.00	1.00	1.00
50K RBF	DDM	176.67	1.00	1.00	0.75	0.75	0.75	0.75
	ECDD	5.00	3.00	37.00	0.03	0.25	0.05	0.08
	HDDM_A	78.00	1.00	0.00	1.00	0.75	0.86	0.87
	HDDM_W	12.34	1.00	30.00	0.09	0.75	0.16	0.26
	STEPD	14.67	1.00	36.00	0.08	0.75	0.14	0.24
	FHDDM	11.34	1.00	11.00	0.21	0.75	0.33	0.40
	RDDM	105.34	1.00	0.00	1.00	0.75	0.86	0.87
	VFDDM	13.00	1.00	22.00	0.12	0.75	0.21	0.30
	SCMER	15.67	1.00	1.00	0.75	0.75	0.75	0.75
100K Agrawal	DDM	69.50	2.00	7.00	0.22	0.50	0.31	0.33
	ECDD	1.50	2.00	243.00	0.01	0.50	0.02	0.06
	HDDM_A	5.25	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	20.75	0.00	6.00	0.40	1.00	0.57	0.63
	STEPD	14.50	0.00	18.00	0.18	1.00	0.31	0.43
	FHDDM	19.00	0.00	0.00	1.00	1.00	1.00	1.00
	RDDM	105.00	0.00	0.00	1.00	1.00	1.00	1.00
	VFDDM	11.75	0.00	3.00	0.57	1.00	0.73	0.76
	SCMER	28.50	0.00	0.00	1.00	1.00	1.00	1.00
100K Asset Negotiation	DDM	165.67	1.00	4.00	0.43	0.75	0.55	0.57
	ECDD	8.00	0.00	116.00	0.03	1.00	0.06	0.18
	HDDM_A	26.75	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	35.00	0.00	0.00	1.00	1.00	1.00	1.00
	STEPD	11.25	0.00	17.00	0.19	1.00	0.32	0.44
	FHDDM	59.25	0.00	0.00	1.00	1.00	1.00	1.00
	RDDM	82.25	0.00	0.00	1.00	1.00	1.00	1.00
	VFDDM	23.25	0.00	136.00	0.03	1.00	0.06	0.17
	SCMER	47.00	0.00	0.00	1.00	1.00	1.00	1.00
100K RBF	DDM	186.34	1.00	4.00	0.43	0.75	0.55	0.57
	ECDD	279.00	3.00	105.00	0.01	0.25	0.02	0.05
	HDDM_A	28.00	1.00	2.00	0.60	0.75	0.67	0.67
	HDDM_W	6.67	1.00	52.00	0.05	0.75	0.10	0.20
	STEPD	59.75	0.00	60.00	0.06	1.00	0.12	0.25
	FHDDM	9.00	1.00	22.00	0.12	0.75	0.21	0.30
	RDDM	113.34	1.00	4.00	0.43	0.75	0.55	0.57
	VFDDM	40.75	0.00	58.00	0.06	1.00	0.12	0.25
	SCMER	15.00	1.00	7.00	0.30	0.75	0.43	0.47

-- table continued on next page

Our method, SCMER, produced results that were equal or reasonably close to ECDD and STEPD in terms of false negatives. SCMER successfully detected all existing drifts in 7 out of the 12 artificial datasets. Its false negative rate was superior or equal to that of DDM. The false positive rate of SCMER was comparable to, and in some cases slightly lower or higher than, HDDM_A-Test and RDDM across most datasets.

Table 7 and Table 8 also includes results for Precision, Recall, F1 and the Matthews Correlation Coefficient (MCC) [36] [37]. Precision measures the proportion of predicted drifts that are actual drifts and is defined as $\frac{TP}{(TP+FP)}$. Recall, given by $\frac{TP}{(TP+FN)}$, is the proportion of existing drifts that were accurately detected. F1 score provides provides the balance between Precision and Recall and is defined as $\frac{2\times Precision\times Recall}{Precision+Recall}$ MCC provides a balanced measure, with larger values indicating better performance; it is calculated as: $\frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)}}$.

Based on these four metrics, the proposed SCMER method achieved the best recall in most scenarios. For F1, MCC and Precision, both SCMER and HDDM_A-Test were the top-performing methods, followed by RDDM and FHDDM. Furthermore, SCMER consistently outperformed DDM, ECDD, STEPD, VFDD, and HDDM_W-Test. From these results, SCMER is a competitive in drift detector for data streams.

Table 8: Concept Drift Identifications in the Abrupt Datasets Using Hoeffding Tree as Base Classifier. (Table Continued from Previous Page)

Dataset	Algorithm	µDist	FN	FP	Precision	Recall	F1	MCC
	DDM	248.25	0.00	4.00	0.50	1.00	0.67	0.71
	ECDD	2.75	0.00	719.00	0.01	1.00	0.01	0.07
	HDDM_A	21.75	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	15.75	0.00	2.00	0.67	1.00	0.80	0.82
300K Agrawal	STEPD	4.25	0.00	44.00	0.08	1.00	0.15	0.29
	FHDDM	18.75	0.00	0.00	1.00	1.00	1.00	1.00
	RDDM	95.50	0.00	2.00	0.67	1.00	0.80	0.82
	VFDDM	4.25	0.00	7.00	0.36	1.00	0.53	0.60
	SCMER	30.75	0.00	0.00	1.00	1.00	1.00	1.00
	DDM	333.25	0.00	3.00	0.57	1.00	0.73	0.76
	ECDD	266.25	0.00	355.00	0.01	1.00	0.02	0.11
	HDDM_A	114.50	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	29.50	0.00	0.00	1.00	1.00	1.00	1.00
300K Asset Negotiation	STEPD	9.75	0.00	41.00	0.09	1.00	0.16	0.30
	FHDDM	58.25	0.00	0.00	1.00	1.00	1.00	1.00
	RDDM	97.75	0.00	4.00	0.50	1.00	0.67	0.71
	VFDDM	26.25	0.00	378.00	0.01	1.00	0.02	0.10
	SCMER	47.75	0.00	0.00	1.00	1.00	1.00	1.00
	DDM	410.34	1.00	2.00	0.60	0.75	0.67	0.67
	ECDD	132.00	0.00	308.00	0.01	1.00	0.03	0.11
	HDDM_A	202.67	1.00	6.00	0.33	0.75	0.46	0.50
	HDDM_W	15.00	1.00	162.00	0.02	0.75	0.04	0.12
300K RBF	STEPD	3.00	3.00	91.00	0.01	0.25	0.02	0.05
	FHDDM	15.34	1.00	68.00	0.04	0.75	0.08	0.18
	RDDM	91.67	1.00	4.00	0.43	0.75	0.55	0.57
	VFDDM	17.34	1.00	141.00	0.02	0.75	0.04	0.13
	SCMER	23.34	1.00	24.00	0.11	0.75	0.19	0.29
	DDM	288.00	0.00	3.00	0.57	1.00	0.73	0.76
	ECDD	184.00	0.00	1126.00	0.00	1.00	0.01	0.06
	HDDM_A	30.75	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	16.00	0.00	2.00	0.67	1.00	0.80	0.82
500K Agrawal	STEPD	5.00	0.00	82.00	0.05	1.00	0.09	0.22
	FHDDM	20.25	0.00	0.00	1.00	1.00	1.00	1.00
	RDDM	105.75	0.00	2.00	0.67	1.00	0.80	0.82
	VFDDM	18.00	0.00	21.00	0.16	1.00	0.28	0.40
	SCMER	24.50	0.00	0.00	1.00	1.00	1.00	1.00
	DDM	367.25	0.00	2.00	0.67	1.00	0.80	0.82
	ECDD	9.75	0.00	609.00	0.01	1.00	0.01	0.08
	HDDM_A	57.75	0.00	0.00	1.00	1.00	1.00	1.00
	HDDM_W	30.00	0.00	0.00	1.00	1.00	1.00	1.00
500K Asset Negotiation	STEPD	9.75	0.00	69.00	0.05	1.00	0.10	0.23
	FHDDM	59.25	0.00	0.00	1.00	1.00	1.00	1.00
	RDDM	97.25	0.00	9.00	0.31	1.00	0.47	0.55
	VFDDM	352.75	0.00	602.00	0.01	1.00	0.01	0.08
	SCMER	42.00	0.00	0.00	1.00	1.00	1.00	1.00
	DDM	441.00	1.00	3.00	0.50	0.75	0.60	0.61
	ECDD	280.00	3.00	568.00	0.00	0.25	0.00	0.02
	HDDM_A	68.00	1.00	9.00	0.25	0.75	0.38	0.43
	HDDM_W	27.67	1.00	267.00	0.01	0.75	0.02	0.09
500K RBF	STEPD	12.34	1.00	150.00	0.02	0.75	0.04	0.12
	FHDDM	12.34	1.00	98.00	0.03	0.75	0.06	0.15
	RDDM	74.34	1.00	4.00	0.43	0.75	0.55	0.57
	VFDDM	7.00	3.00	226.00	0.00	0.25	0.01	0.03
	SCMER	19.67	1.00	35.00	0.08	0.75	0.14	0.24

5.3. Runtime and memory analysis

Tables 9, 10 provide runtime usage for all the methods tested and tables 11, 12 shows the memory usage for all the methods. For the runtime and memory analysis, we used the Bonferroni-Dunn post-hoc test. The results of the tests are presented graphically in Figures 12, 13, 14 and 15.

In case of runtime for Naive Bayes as classifier, we can see that SCMER, FHDDM, RDDM, HDDM_A and DDM were having the fastest runtimes with no reliable runtime differences among them. VFDD, HDDM_W, STEPD and ECDD were having the slowest runtimes. In that, ECDD has 2 times slower than every other method. Similarly, for Hoeffding Tree as classifier, we can see the same kind of results, with ECDD having the slowest runtimes. Hence, from the statistical tests we can say that SCMER is among the fastest, with similar runtimes of FHDDM, RDDM, HDDM_A and DDM.

For the memory usage, SCMER uses the least memory of all the other methods on all the 24 datasets with no exceptions. Other than SCMER, the statistically closest methods with least memory usage are ECDD, FHDDM and DDM. The methods which are not statistically worst are HDDM_A, STEPD, HDDM_W, VFDD and RDDM. Hence from the results, SCMER uses least memory for detection of drifts in the data streams.

Considering the experimental outcomes for accuracy, drift identification, runtime and memory analysis, we conclude that SCMER is the best-performing method overall.

Table 9: Runtime for Drift Detectors Using Naive Bayes in the Artificial Datasets

Type-Size	Abrupt-50K			Abrupt-100K			Abrupt-300K			Abrupt-500K		
Dataset	Agrawal	Asset	RBF	Agrawal	Asset	RBF	Agrawal	Asset	RBF	Agrawal	Asset	RBF
DDM	0.2681	0.1697	0.4366	0.4269	0.2627	0.8131	1.0071	0.5533	2.2486	1.5893	0.8080	3.6382
ECDD	0.3104	0.1899	0.4954	0.4952	0.3067	0.8978	1.1178	0.6371	2.3924	1.8101	0.9450	3.7254
HDDM_A	0.2884	0.1620	0.4425	0.4467	0.2603	0.7997	1.0348	0.5467	2.2416	1.6120	0.8501	3.6672
HDDM_W	0.3229	0.1652	0.4600	0.4398	0.2737	0.8507	1.0511	0.5500	2.2693	1.6434	0.8395	3.7047
STEPD	0.3022	0.1772	0.4465	0.4301	0.2781	0.8257	1.0345	0.5543	2.3367	1.6401	0.8543	3.6598
FHDDM	0.2730	0.1616	0.4318	0.4527	0.2615	0.8034	1.0338	0.5235	2.2688	1.6193	0.7917	3.6140
RDDM	0.2526	0.1650	0.4331	0.4145	0.2662	0.7901	1.0141	0.5735	2.2455	1.5943	0.8430	3.6183
VFDD	0.2923	0.1959	0.4587	0.4551	0.3243	0.8292	1.0835	0.6715	2.3373	1.7091	1.0162	3.8335
SCMER	0.2843	0.1613	0.4348	0.4245	0.2591	0.7990	1.0085	0.5216	2.2236	1.5929	0.7908	3.5333
Type-Size	Gradual-50K			Gradual-100K			Gradual-300K			Gradual-500K		
Dataset	Agrawal	Asset	RBF	Agrawal	Asset	RBF	Agrawal	Asset	RBF	Agrawal	Asset	RBF
DDM	0.2670	0.1624	0.5463	0.4493	0.2598	0.9829	1.0055	0.5420	2.7453	1.5770	0.8009	3.7292
ECDD	0.2910	0.1858	0.5046	0.4951	0.3035	0.8761	1.1305	0.6369	2.4477	1.7638	0.9420	3.7906
HDDM_A	0.2609	0.1626	0.5575	0.4361	0.2661	0.9035	1.0072	0.5558	2.3458	1.6120	0.8441	3.7286
HDDM_W	0.2942	0.1659	0.4672	0.4337	0.2697	0.8297	1.0559	0.5446	2.3042	1.6432	0.8446	3.6560
STEPD	0.2679	0.1710	0.4611	0.4267	0.2729	0.8316	1.0536	0.5569	2.2783	1.7588	0.8619	3.7208
FHDDM	0.2683	0.1643	0.4621	0.4693	0.2595	0.8264	1.0329	0.5231	2.3430	1.7647	0.7999	3.6492
RDDM	0.2607	0.1718	0.5389	0.4107	0.2662	1.0182	1.0297	0.5338	2.3604	1.6241	0.8307	3.7772
VFDD	0.3363	0.2020	0.4773	0.4453	0.3147	0.8494	1.0819	0.6625	2.3475	1.7433	0.9603	3.6543
SCMER	0.2922	0.1588	0.4759	0.4173	0.2516	0.8368	0.9914	0.5159	2.3059	1.6019	0.7917	3.6675

Table 10: Runtime for Drift Detectors Using Hoeffding Tree in the Artificial Datasets

Type-Size	Abrupt-50K			Abrupt-100K			Abrupt-300K			Abrupt-500K		
Dataset	Agrawal	Asset	RBF	Agrawal	Asset	RBF	Agrawal	Asset	RBF	Agrawal	Asset	RBF
DDM	0.3232	0.2374	0.6345	0.4869	0.3402	0.8109	1.0524	0.8036	2.2620	1.6430	1.0930	3.6870
ECDD	0.6219	0.3682	1.1508	1.0149	0.5207	1.4408	2.4911	1.1389	4.1921	4.1018	1.7412	6.6391
HDDM_A	0.3194	0.2292	0.7532	0.4735	0.3493	0.8012	1.0477	0.7315	2.2673	1.7378	1.1430	3.2102
HDDM_W	0.3433	0.2263	0.6235	0.5079	0.3497	0.8474	1.0925	0.7396	2.3547	1.6677	1.1162	3.5581
STEPD	0.3578	0.2372	0.6181	0.5751	0.3611	0.9924	1.1958	0.7205	2.6191	1.9028	1.0724	4.0537
FHDDM	0.3172	0.2325	0.6907	0.4797	0.3543	0.8058	1.0470	0.7067	2.0663	1.6237	1.1115	3.2325
RDDM	0.3185	0.2403	0.6315	0.4726	0.3449	0.8165	1.0415	0.7204	2.2040	1.6147	1.0626	3.3728
VFDD	0.3164	0.2450	0.6933	0.4843	0.3991	0.8112	1.0791	0.7866	2.1593	1.6584	1.1538	3.3195
SCMER	0.3182	0.2218	0.6129	0.4741	0.3344	0.7935	1.0416	0.7015	2.1177	1.6824	1.1114	3.3430
Type-Size	Gradual-50K			Gradual-100K			Gradual-300K			Gradual-500K		
Dataset	Agrawal	Asset	RBF	Agrawal	Asset	RBF	Agrawal	Asset	RBF	Agrawal	Asset	RBF
DDM	0.3219	0.2422	0.5157	0.4881	0.3454	1.0640	1.0505	0.7422	2.1479	1.5867	1.1031	3.3573
ECDD	0.6333	0.3379	0.8166	1.0065	0.5642	1.5027	2.4601	1.1320	4.1108	4.0835	1.8279	6.7274
HDDM_A	0.3195	0.2328	0.5161	0.4805	0.4093	0.8643	1.0547	0.7287	2.1460	1.6631	1.1399	3.3374
HDDM_W	0.3322	0.2321	0.5309	0.4983	0.3930	0.8669	1.0765	0.7270	2.2853	1.6807	1.1199	3.5829
STEPD	0.3689	0.2379	0.5805	0.5747	0.4082	0.9592	1.2097	0.7157	2.5245	1.9720	1.0782	4.0225
FHDDM	0.3200	0.2337	0.4864	0.4767	0.3524	0.8346	1.0476	0.7223	2.0546	1.6270	1.0997	3.1821
RDDM	0.3072	0.2293	0.5144	0.4740	0.3825	1.0267	1.1106	0.7211	2.4419	1.6025	1.0531	3.9881
VFDD	0.3162	0.2417	0.5122	0.4894	0.3735	0.8291	1.0772	0.7716	2.2937	1.6728	1.1477	3.3240
SCMER	0.3278	0.2237	0.4977	0.4694	0.3418	0.8090	1.0295	0.7227	2.1603	1.6990	1.1233	3.2060

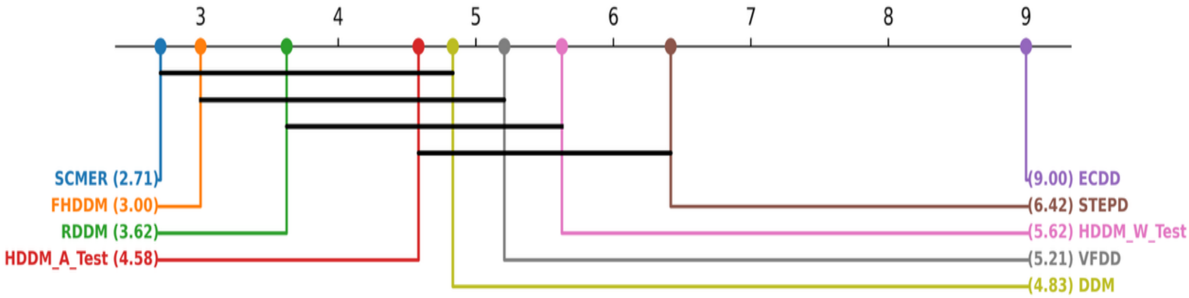


Fig. 12: Comparison Results of Runtime Using the Bonferroni Post-Hoc Test on All Tested Datasets for Naive Bayes Classifier.

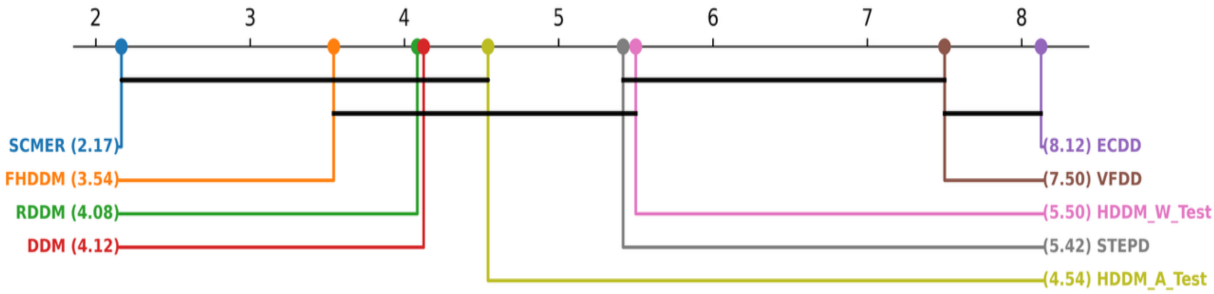


Fig. 13: Comparison Results of Runtime Using the Bonferroni Post-Hoc Test on All Tested Datasets for Hoeffding Tree Classifier.

Table 11: Memory Analysis for Drift Detectors Using Naive Bayes in the Artificial Datasets

Type-Size	Abrupt-50K			Abrupt-100K			Abrupt-300K			Abrupt-500K		
Dataset	Agrawal	Asset	RBF	Agrawal	Asset	RBF	Agrawal	Asset	RBF	Agrawal	Asset	RBF
DDM	2.8025	2.8042	2.6860	2.8020	2.8096	2.6786	2.8041	2.8050	2.6902	2.8044	2.8061	2.6728
ECDD	2.6135	2.6181	2.5055	2.6127	2.6192	2.5012	2.6114	2.6194	2.5026	2.6112	2.6197	2.4984
HDDM_A	2.8984	2.9063	2.7796	2.8984	2.9063	2.7807	2.8997	2.9071	2.7993	2.9021	2.9076	2.7911
HDDM_W	3.0579	3.0625	2.9416	3.0587	3.0637	2.9413	3.0588	3.0658	2.9469	3.0592	3.0663	2.9408
STEPD	2.9097	2.9201	2.7997	2.9093	2.9199	2.8006	2.9107	2.9193	2.8119	2.9104	2.9193	2.8040
FHDDM	2.7891	2.7969	2.6624	2.7891	2.7969	2.6597	2.7891	2.7969	2.6633	2.7891	2.7969	2.6586
RDDM	10.1485	10.1545	10.0520	10.1491	10.1574	10.0443	10.1504	10.1585	10.0429	10.1478	10.1592	10.0460
VFDD	3.7578	3.7656	3.6312	3.7578	3.7656	3.6263	3.7578	3.7656	3.6310	3.7578	3.7656	3.6272
SCMER	2.5859	2.5938	2.4584	2.5859	2.5938	2.4565	2.5859	2.5938	2.4601	2.5859	2.5938	2.4555
Type-Size	Gradual-50K			Gradual-100K			Gradual-300K			Gradual-500K		
Dataset	Agrawal	Asset	RBF	Agrawal	Asset	RBF	Agrawal	Asset	RBF	Agrawal	Asset	RBF
DDM	2.8020	2.8043	2.7008	2.8029	2.8090	2.7027	2.8041	2.8111	2.7215	2.8030	2.8082	2.7131
ECDD	2.6130	2.6172	2.5069	2.6116	2.6189	2.5017	2.6115	2.6196	2.5066	2.6111	2.6197	2.4981
HDDM_A	2.8990	2.9063	2.8144	2.8985	2.9063	2.7951	2.8997	2.9072	2.8031	2.9021	2.9077	2.7933
HDDM_W	3.0592	3.0625	2.9448	3.0594	3.0637	2.9417	3.0590	3.0659	2.9466	3.0591	3.0663	2.9412
STEPD	2.9085	2.9204	2.8011	2.9088	2.9203	2.8033	2.9106	2.9190	2.8116	2.9103	2.9194	2.8031
FHDDM	2.7891	2.7969	2.6689	2.7891	2.7969	2.6635	2.7891	2.7969	2.6632	2.7891	2.7969	2.6591
RDDM	10.1501	10.1579	10.0625	10.1497	10.1571	10.0546	10.1501	10.1604	10.0585	10.1488	10.1604	10.0516
VFDD	3.7578	3.7656	3.6347	3.7578	3.7656	3.6299	3.7578	3.7656	3.6314	3.7578	3.7656	3.6283
SCMER	2.5859	2.5938	2.4705	2.5859	2.5938	2.4628	2.5859	2.5938	2.4626	2.5859	2.5938	2.4573

Table 12: Memory Analysis for Drift Detectors Using Hoeffding Tree in the Artificial Datasets

Type-Size	Abrupt-50K			Abrupt-100K			Abrupt-300K			Abrupt-500K		
Dataset	Agrawal	Asset	RBF	Agrawal	Dataset	Agrawal	Asset	RBF	Agrawal	Dataset	Agrawal	Asset
DDM	8.5376	8.5078	8.3834	8.5691	8.5049	8.3491	8.5696	8.4546	8.3882	8.5698	8.4598	8.3519
ECDD	8.3476	8.3388	8.2514	8.3410	8.3483	8.2429	8.3469	8.3430	8.2328	8.3493	8.3450	8.2255
HDDM_A	8.5623	8.5087	8.4589	8.5838	8.5084	8.4732	8.6215	8.5080	8.5217	8.6561	8.5078	8.5069
HDDM_W	8.7656	8.6653	8.6115	8.7602	8.6647	8.6238	8.7874	8.6642	8.6350	8.8180	8.6641	8.6294
STEPD	8.6469	8.6745	8.5271	8.6445	8.6644	8.5413	8.6623	8.6665	8.5651	8.6738	8.6695	8.5567
FHDDM	8.4053	8.3989	8.2728	8.4057	8.3990	8.2723	8.4060	8.3986	8.2762	8.4063	8.3984	8.2720
RDDM	15.8680	15.7922	15.6845	15.8146	15.7881	15.7396	15.8874	15.7502	15.7747	15.9219	15.7654	15.7572
VFDD	9.3727	9.3732	9.2413	9.3735	9.3735	9.2379	9.3741	9.3718	9.2433	9.3742	9.3714	9.2394
SCMER	8.2022	8.1962	8.0674	8.2025	8.1958	8.0689	8.2029	8.1955	8.0730	8.2031	8.1953	8.0700
Type-Size	Gradual-50K			Gradual-100K			Gradual-300K			Gradual-500K		
Dataset	Agrawal	Asset	RBF	Agrawal	Dataset	Agrawal	Asset	RBF	Agrawal	Dataset	Agrawal	Asset
DDM	8.5607	8.5077	8.4149	8.5691	8.5066	8.4054	8.5696	8.4853	8.4694	8.5698	8.5186	8.4742
ECDD	8.3521	8.3334	8.2517	8.3429	8.3433	8.2359	8.3476	8.3464	8.2356	8.3488	8.3455	8.2252
HDDM_A	8.5922	8.5087	8.5115	8.5975	8.5083	8.4722	8.6207	8.5080	8.4683	8.6553	8.5079	8.4710
HDDM_W	8.7653	8.6651	8.6121	8.7654	8.6646	8.6202	8.7877	8.6642	8.6355	8.8153	8.6642	8.6301
STEPD	8.6527	8.6698	8.5250	8.6486	8.6665	8.5430	8.6646	8.6670	8.5670	8.6726	8.6648	8.5563
FHDDM	8.4053	8.3989	8.2787	8.4057	8.3989	8.2757	8.4061	8.3986	8.2769	8.4063	8.3985	8.2733
RDDM	15.8226	15.7911	15.7107	15.8382	15.7877	15.7097	15.8872	15.8233	15.7726	15.9219	15.8469	15.7822
VFDD	9.3729	9.3730	9.2451	9.3738	9.3731	9.2416	9.3742	9.3732	9.2438	9.3742	9.3733	9.2396
SCMER	8.2022	8.1957	8.0786	8.2025	8.1958	8.0721	8.2029	8.1955	8.0754	8.2031	8.1954	8.0701

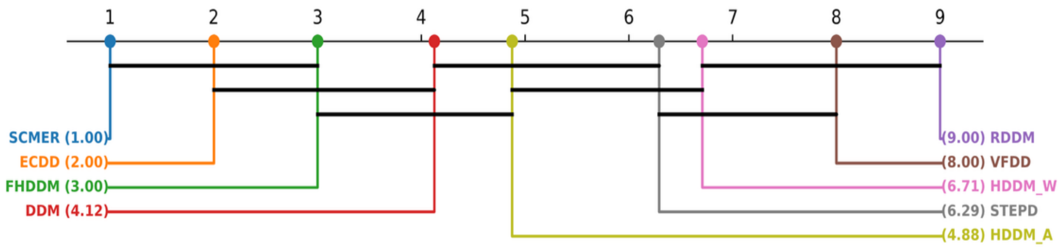


Fig. 14: Comparison Results of Memory Using the Bonferroni Post-Hoc Test on All Tested Datasets for Naive Bayes Classifier.

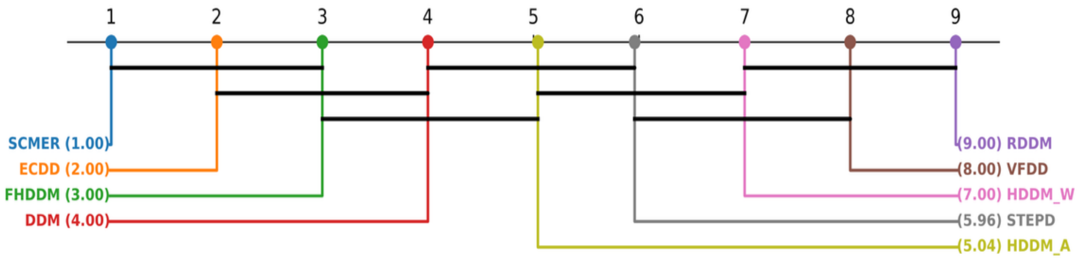


Fig. 15: Comparison Results of Memory Using the Bonferroni Post-Hoc Test on All Tested Datasets for Hoeffding Tree Classifier.

6. Conclusion

This article presented a new approach, SCMER, for detecting concept drifts in data streams. The proposed method introduces a novel quantification for drift detection based on a shifting classifier's mean error rate. To detect distributional changes, SCMER uses the Hellinger distance inspired metric and Hoeffding's bound to estimate drift positions.

Experiments were conducted to evaluate SCMER against existing methods, namely DDM, ECDD, STEPD, FHDDM, HDDM_A-Test, HDDM_W-Test, VFDD and RDDM. A Naive Bayes (NB) classifier and Hoeffding Tree (HT) classifier were used as the base learner, and the evaluation utilized datasets of different sizes generated by three distinct data generators, as well as real-world datasets.

The experimental results, presented in Table 5 and Table 6, show that SCMER performed comparably to or better than the other methods on most datasets. Furthermore, SCMER's accuracy was slightly higher on datasets with abrupt drifts than on those with gradual drifts for NB and opposite in case of HT, which indicate SCMER performed well in both of the scenarios.

In terms of drift identification, SCMER produced excellent results with low false positives (FP), low false negatives (FN), and the smallest mean distance to the true drift locations. In contrast, the performance of ECDD and STEPD was among the worst on most datasets. In terms of memory analysis, SCMER used least memory than any other tested methods which makes our approach to be suitable for drift detection.

Based on these test outcomes, we assert that SCMER is the best-performing drift detection method in this study, with HDDM_A-Test, FHDDM and RDDM following closely behind.

For future work, we plan to address other challenges associated with concept drift in data streams also historical prediction results are not preserved in our approach due to this we are avoiding recurring concepts which can be addressed in the future.

References

[1] I. Khamassi, M. Sayed-Mouchaweh, M. Hammami, and K. Ghédira, "Discussion and review on evolving data streams and concept drift adapting," *Evolving Systems*, 2018, https://doi.org/10.1007/s12530-016-9168-2.

[2] Y. K. Lee, L. Wang, and K. H. Ryu, "A system architecture for monitoring sensor data stream," in *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, Aizu-Wakamatsu, Fukushima, Japan: IEEE, Oct. 2007, pp. 1026–1031. https://doi.org/10.1109/CIT.2007.178.

[3] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Tracking recurring contexts using ensemble classifiers: An application to email filtering," *Knowl. Inf. Syst.*, vol. 22, no. 3, 2010, https://doi.org/10.1007/s10115-009-0206-2.

[4] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič, "Stream-based active learning for sentiment analysis in the financial domain," *Inf. Sci. (N. Y).*, vol. 285, pp. 181–203, Nov. 2014, https://doi.org/10.1016/j.ins.2014.04.034.

[5] I. Žliobaitė, M. Pechenizkiy, and J. Gama, "An overview of concept drift applications," in *Big Data Analysis: New Algorithms for a New Society*, vol. 16, Cham: Springer International Publishing, 2016, pp. 91–114. https://doi.org/10.1007/978-3-319-26989-4_4.

[6] J. Gama, *Knowledge discovery from data streams*. CRC Press, 2010. https://doi.org/10.1201/EBK1439826119.

[7] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive Online Analysis," *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010, [Online]. Available: http://portal.acm.org/citation.cfm?id=1859903.

[8] J. Gama, P. Medas, G. Castillo, and P. P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence – SBIA 2004*, vol. 3171, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 286–295. https://doi.org/10.1007/978-3-540-28645-5_29.

[9] M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, "Early drift detection method," in *Fourth international workshop on knowledge discovery from data streams*, 2006, pp. 77–86.

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

[11] K. Nishida and K. Yamauchi, "Detecting concept drift using statistical testing," in *International conference on discovery science*, 2007, pp. 264–269. https://doi.org/10.1007/978-3-540-75488-6_27.

[12] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, p. 100, Jun. 1954, https://doi.org/10.2307/2333009.

[13] S. H. Bach and M. A. Maloof, "Paired learners for concept drift," in *2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italy: IEEE, Dec. 2008, pp. 23–32. https://doi.org/10.1109/ICDM.2008.119.

[14] G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand, "Exponentially weighted moving average charts for detecting concept drift," *Pattern Recognit. Lett.*, vol. 33, no. 2, pp. 191–198, Jan. 2012, https://doi.org/10.1016/j.patrec.2011.08.019.

[15] I. Fr\'\ias-Blanco, J. del Campo-Ávila, G. Ramos-Jiménez, R. Morales-Bueno, A. Ortiz-D\'\iaz, and Y. Caballero-Mota, "Online and non-parametric drift detection methods based on Hoeffding's bounds," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 810–823, 2015. https://doi.org/10.1109/TKDE.2014.2345382.

[16] A. Pesaranghader and H. L. Viktor, "Fast hoeffding drift detection method for evolving data streams," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[17] 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.*, 2017, https://doi.org/10.1016/j.eswa.2017.08.023.

[18] M. Han, F. Meng, and C. Li, "Variance Feedback Drift Detection Method for Evolving Data Streams Mining," *Applied Sciences*, vol. 14, no. 16, 2024, https://doi.org/10.3390/app14167157.

[19] T. M. Mitchell and others, "Machine learning. 1997," *Burr Ridge, IL: McGraw Hill*, vol. 45, no. 37, pp. 870–877, 1997.

[20] J. S. Hunter, "The exponentially weighted moving average," *Journal of Quality Technology*, vol. 18, no. 4, pp. 203–210, Oct. 1986, https://doi.org/10.1080/00224065.1986.11979014.

[21] D. R. de L. Cabral and R. S. M. de Barros, "Concept drift detection based on Fisher's Exact test," *Inf. Sci. (N. Y).*, vol. 442–443, pp. 220–234, 2018, https://doi.org/10.1016/j.ins.2018.02.054.

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

[23] X. Yu, L. Ruan, J. S. Evans, and E. Wong, "Adaptive windowing-based concept drift detection and adaptation framework for human-to-machine applications over future communication networks," *J. Opt. Commun. Netw.*, vol. 17, no. 4, pp. 338–351, Apr. 2025, https://doi.org/10.1364/JOCN.538964.

[24] H. Li and T. Zhao, "A dynamic similarity weighted evolving fuzzy system for concept drift of data streams," *Inf. Sci. (N. Y).*, vol. 659, p. 120062, 2024, https://doi.org/10.1016/j.ins.2023.120062.

[25] L. Hu, Y. Lu, and Y. Feng, "Concept Drift Detection Based on Deep Neural Networks and Autoencoders," *Applied Sciences*, vol. 15, no. 6, 2025, https://doi.org/10.3390/app15063056.

[26] B. I. F. Maciel, S. G. T. C. Santos, and R. S. M. Barros, "A Lightweight Concept Drift Detection Ensemble," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, Vietri sul Mare, Italy: IEEE, Nov. 2015, pp. 1061–1068. https://doi.org/10.1109/ICTAI.2015.151.

[27] F. Enembreck, B. C. Ávila, E. E. Scalabrin, and J.-P. Barthès, "Learning drifting negotiations," *Applied Artificial Intelligence*, vol. 21, no. 9, pp. 861–881, 2007. https://doi.org/10.1080/08839510701526954.

[28] J. Gama, R. Sebastião, and P. P. Rodrigues, "On evaluating stream learning algorithms," *Mach. Learn.*, vol. 90, no. 3, pp. 317–346, Mar. 2013, https://doi.org/10.1007/s10994-012-5320-9.

[29] A. Bifet, J. Read, I. Zliobaite, B. Pfahringer, and G. Holmes, "Pitfalls in benchmarking data stream classification and how to avoid them," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2013, pp. 465–479. https://doi.org/10.1007/978-3-642-40988-2_30.

[30] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009. https://doi.org/10.1145/1557019.1557041.

[31] C. O. Sakar, S. O. Polat, M. Katircioglu, and Y. Kastro, "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks," *Neural Comput. Appl.*, vol. 31, no. 10, pp. 6893–6908, 2019. https://doi.org/10.1007/s00521-018-3523-0.

[32] A. V Uzilov, J. M. Keegan, and D. H. Mathews, "Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change," *BMC Bioinformatics*, vol. 7, no. 1, p. 173, 2006, https://doi.org/10.1186/1471-2105-7-173.

[33] B. Strack *et al.*, "Impact of HbA1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records," *Biomed Res. Int.*, vol. 2014, 2014, https://doi.org/10.1155/2014/781670.

[34] R. Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid," *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, vol. 7, no. Utgo 1988, 1996.

[35] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[36] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, 2006, https://doi.org/10.1016/j.patrec.2005.10.010.

[37] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *BBA - Protein Structure*, vol. 405, no. 2, 1975, https://doi.org/10.1016/0005-2795(75)90109-9.