Handling the Concept Drifts Based on Ensemble Learning with Adaptive Windows

Tin Mar Myint and Khin Thidar Lynn

***Abstract*—Continuous learning from streaming data is one of the contemporary most challenging topics. Learning algorithms not only need to handle fast-moving big data but should also be able to adapt to future evolving changes. The evolving structure of data streams is the phenomenon referred to as concept drift. Two types of concept drift handling based on ensemble approaches are online ensemble and chunk-based ensemble (window-based ensemble). The main disadvantage of the chunk-based ensembles is the difficulty of tuning the block size to provide a trade-off between fast reactions to drifts. In this paper, we propose Accuracy Updated Ensemble-2 (AUE2) based on the adaptive windowing approach (denoted as A-AUE2) by using Brier Skill Score as sudden and gradual changes detector. Moreover, K-Nearest Neighbors (KNN) based noise filtering method is applied to eliminate the noisy samples from each of the adaptive windows to improve the effectiveness of the ensemble learning approach. This paper's proposed approach has been evaluated on six artificial data sets and four real benchmark data sets by using two base learners. Specially, we illustrate that the proposed approach has outperformed the other state-of-the-art drift detection and handling approaches by using performance metrics and statistical analysis.**

***Index Terms*—Concept Drift, Data Stream, Adaptive Window, Ensemble Learning, Drift Detection, Noise.**

I. Introduction

THE three main challenges of data stream mining are speed, size and variability [14], [37]-[38]. Speed and size depend on time and limited memory, and they cause the algorithms to temporarily store the incoming data and to process them only once. Variability in the data stream refers to the changing data stream over time, which in unpredictable ways influences the underlying concept of the data stream over time, and it is the most important challenge in the real-world application. The phenomenon of the evolving data stream over time is called concept drift. The real-life application areas of concept drifts are spam mail detection, customer preferences, weather forecasting, financial transactions and credit card fraud detection [2], [11], [16]. For example, weather prediction rules can change radically with the season [30]. Moreover, customer preferences may change over time, depending on current weekdays, availability of alternatives, inflation, etc. The cause of change is often hidden that is not known a priori. It makes the learning task more complicated. So, the concept drift is an important challenge if the incoming data streams are not stationary. The underlying changes in the distribution may be sudden or abrupt drift, gradual drift, incremental drift and recurrent drift [1], [11], [16]. Fig. 1 presents the nature of different types of concept drift.

(1) Sudden or abrupt drift: Sudden or abrupt drift is a quick replacement from one concept to another. Sudden drift can degrade the performance of the model directly and can be easily detected by the detection methods.

(2) Gradual drift: The gradual drift refers to a phase of transition in which instances of two different distributions are mixed.

(3) Incremental drift: Incremental drift is a unique case of gradual drift when the concept is derived from more than two sources and the possibility that each concept will evolve. Moreover, the gradual and incremental drifts are more challenging to detect than sudden drift due to their small change rates and overlapping data distributions.

(4) Recurrent drift: The recurring context is the representation of a concept after the passage of time.

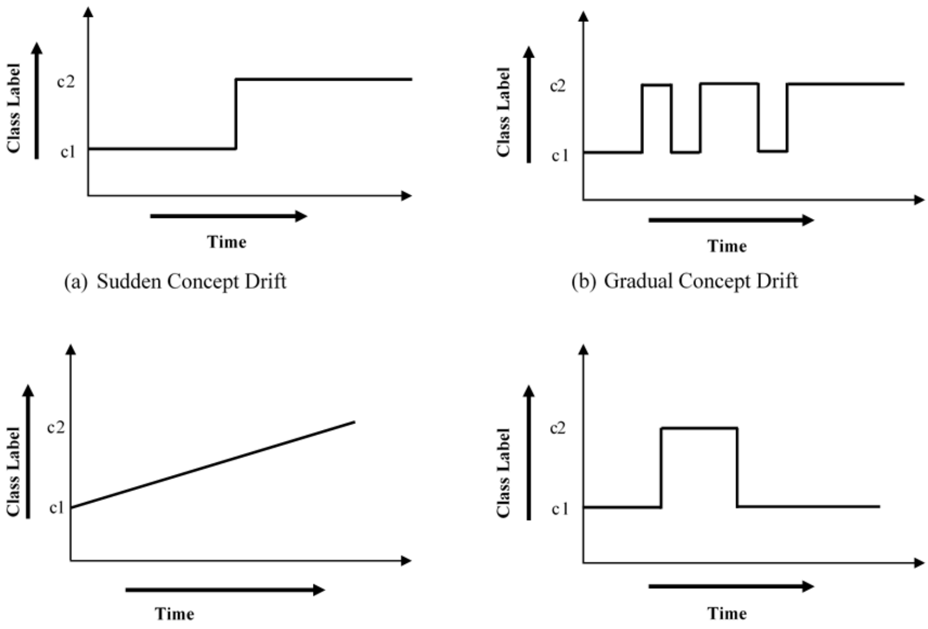
The problem of concept drift has been considered to mine data with optimal accuracy level. Data stream learning is rapidly attracted to research due to many difficulties in the streaming of the real-world. Concept drift detection is a problem in these difficulties of data steam where the distribution of data changes and the current prediction model becomes inaccurate or ineffective. Learners of the data stream are generally classified into single and ensemble classifiers. Ensemble learning is the combination of single classifiers whose decisions are aggregated by a voting rule. The prediction of ensemble learning is more accurate than a single classifier because it combines the decision of many single classifiers. Ensemble learning is a popular method for improving the accuracy of static analysis classification problems but they need to be generalized for changing environments. The generalization of ensemble learning may include the changing of ensemble structure, updating the technique of integration, or introducing direct online learning from a single incoming sample [12], [18], [33].

For detecting the concept drifts, many heuristic approaches have been proposed based on the two approaches of the ensembles: (1) online ensemble and (2) chunk-based ensemble. The online ensemble approach can react quickly to abrupt drifts through one-by-one processing. But, the online ensemble approach incurs high computational costs due to frequent updating of the model. The chunk-based ensemble approach adapts the concept drifts by creating new component classifiers from new chunks of training samples. The main drawback of the chunk-based ensemble is the difficultly of tuning the chunk sizes or window sizes. Accuracy Updated Ensemble-2 (AUE2) is proposed by integrating the accuracy-based weighting mechanisms with the incremental nature of Hoeffding Trees as a base learner [18]. AUE2 is the fixed-size chunk-based ensembles approach and AUE2 incrementally updates every previous member of the

Manuscript received September 20, 2020; revised April 29, 2021.

Tin Mar Myint is a PhD candidate at Data Mining and Machine Learning Lab, University of Computer Studies, Mandalay, Myanmar, E-mail: tinmarmyint@ucsm.edu.mm

Khin Thidar Lynn is a Professor at Faculty of Information Science, University of Computer Studies, Mandalay, Myanmar, E-mail: khinthidalynn@ucsm.edu.mm



(a) Sudden Concept Drift

(b) Gradual Concept Drift

(c) Incremental Concept Drift

(d) Recurrent Concept Drift

Fig. 1. Types of Concept Drift

ensemble with observations in the most recent blocks. AUE2 intends to respond equally well to different concept drifts.

In this paper, the proposed A-AUE2 approach focuses on the adaptive ensembles that sequentially create the component classifiers from the adaptive windows. In handling the different types of concept drift, A-AUE2 is less dependent on chunk sizes than AUE2 because it uses the adaptive windows based on Brier Skill Score method. Furthermore, a difficult issue in the handling of concept drifts is to discriminate the concept drifts and noise. The processing of noisy data in the data streams is one of the most critical areas. In this paper, A-AUE2 handles the problem of distinguishing between concept drift and noise because of filtering the noisy samples from each of the adaptive windows or chunks by using the K-Nearest Neighbors (KNN). Our proposed A-AUE2 approach is better than AUE2 for different types of concept drift in artificial and real data streams with Hoeffding Tree (HT) and Hoeffding Adaptive Tree that has a classifier at the leaves (denoted as HATCL), respectively, as base learner. And, the effectiveness of our proposed approach is validated with Precision, Recall, F-Measure (F1), Matthews Correlation Coefficient (MCC) and Area Under the (Receiver Operating Characteristics (ROC)) Curve (AUC) and statistically significant tests.

The main contributions of this paper are as follows:

1) First, the different windows are specified for each ensemble member by using the adaptive windowing method, as sudden and gradual changes detector in the data stream to support the tackling of four types of concept drift.
2) Second, we handle the noisy samples over each of the adaptive sliding windows to improve the effectiveness of the ensemble learning approach.

The remainder of this paper is arranged accordingly: Section 2 includes related work on concept drift and the concept drifts' handling mechanisms. The detail of the proposed methodology is described in section 3. Section 4 includes the experimental setup of the proposed approach. Section 5 illustrates the results, discussions and comparisons of the proposed approach with other methods. Finally, the conclusion and future work are presented in Section 6.

II. Related Works

A. Concept Drift

Concept drift refers to the change in the relation of input and output data over time in the underlying problem. Concept drift takes place when the concept in which data are collected changes from time to time after a minimum period of stability [2]. At each point in time t, every instance is created by a source with a joint probability distribution $p^t(x+y)$. If all instances are generated by the same distribution, data concepts are stable or stationary. If there exists x such that two distinct points $p^t(x+y) \neq p^{t+\triangle}(x+y)$ in time t and $t+\triangle$, concept drift has occurred. Different components of $p^t(x+y)$ may change [36]. Concept drift detection methods have two categories:

1) Single Classifier for Concept Drift Detection and Handling
2) Ensemble Classifiers for Concept Drift Detection and Handling

1) Single Classifier for Concept Drift Detection and Handling: In data stream environments, the concept drift methods analyze the predictions from the basic classifier and apply a decision model to detect changes in the data distribution. Concept drift detection methods monitor the performance of the base classifier and decide when the concept drifts have occurred. In detection methods, a lower confidence level is used for indicating the warning and signaling the drift.

Drift Detection Method (DDM) detects concept drifts from a set of samples by analyzing the probability of error and standard deviation [4]. DDM's parameters include the minimum number of instances, the value of warning and drift. DDM detects sudden and significant changes well but is unable to detect gradual and small changes. Early Drift Detection Method (EDDM) is like DDM, which uses the difference between two errors in classification [5]. The parameters of EDDM include the value of warning, the value of drift and the minimum number of errors before the drift detects. EDDM is effective on gradual drift but noise sensitive.

Statistical Test of Equal Proportions (STEPD) is similar to DDM and EDDM but STEPD uses the accuracy of the base classifier over two windows [6]. STEPD method tracks an online classifier's predictive accuracy. The core idea of STEPD is to consider two online classified predictive accuracy: (1) the recent accuracy and (2) the overall accuracy. STEPD suggests a statistical significance after the concept drift has occurred between the accuracy for recent samples and the overall accuracy from the beginning of the learning. When detecting sudden changes, STEPD is much faster and more efficient than DDM.

Error Distance Approach for Drift Detection and Monitoring (EDIST2) detects concept drifts through two windows [7]. EDIST2 uses the number of errors over two windows and the distance between the errors. Concept-Adapting Very Fast Decision Tree (CVFDT) extends Very Fast Decision Tree (VFDT) to illustrate the concept drifts with a fixed sliding window [8]. Fast-DBSCAN (FDBSCAN) can filter noise and detect concept drifts [9]. FDBSCAN algorithm handles noise data and classification algorithm (FDBCA) detects concept drifting with Mean Square Error (MSE). Fisher Proportions Drift Detector (FPDD), Fisher Square Drift Detector (FSDD) and Fisher Test Drift Detector (FTDD) detect drifts with Fisher's Exact Test by monitoring the number of errors or correct predictions in each of the two windows [10].

Wilcoxon Rank Sum Test Drift Detector (WSTD) monitors the predictions of the base learner using two windows. WSTD calculates the ranks and P_{value} using the predictions of two windows to detect sudden and gradual drifts [44].

2) Ensemble Classifiers for Concept Drift Detection and Handling: Knowledge-Maximized Ensemble (KME) is integrated the chunk-based and online ensembles methods for dealing with various forms of concept drift [11]. KME can be treated as a model of a hybrid ensemble. In KME, abrupt drifts are observed in the drift detection method and KME is an efficient approach based on the component evaluation and the weighting methods for the three different drifts.

Accuracy Weighted Ensemble (AWE) is a powerful chunks-based ensemble by training classification elements on consecutive data pieces and incorporating the current chunks to evaluate all other components [12]. In the final vote of AWE, several of the best members are selected. The weight of each component from AWE focuses on the mean square error (MSE) based on the latest chunk observations.

Moreover, Krawczyk [13] proposed a lightweight improvement that can be used to improve their strength to drifts and noise on any online ensemble method. This method focuses on uncertain classifiers to abstain from making a prediction that is especially useful for noisy data streams. Moreover, Entropy-Based Ensemble (EBE) is proposed the incorporation of information entropy to detect concept drifts in the evolving ensemble [14]. EBE compared the entropies of two blocks for detecting concept changes.

Besides, Jadhav [15] proposed the single ensemble by combining the online and block-based classifiers for sudden and gradual drift . This method handles the missing value of attributes. In this method, the class label is predicted and updated instantaneously when the data instance is accessible. Then, the instance is stored in the buffer. When the buffer is complete, block-based classifiers use blocks of buffer for class predictions and block-based classifiers that are modified for these block instances.

Number and Distance of Errors (NDE) is introduced as a novel explicit approach for detecting concept drifts based on ensemble learners [16]. NDE method processes sample one by one and monitors the error distribution of the ensemble to detect probable drifts. The new concept of the NDE method is trained to keep the model up to date when the drift is detected.

Multi-Label Ensemble with Adaptive Windowing (MLAW) demonstrates a new identification of changes based on Jensen-Shannon divergences to identify different types of concept drift in data streams [17]. MLAW measures the distribution between two sequential windows using Jensen-Shannon divergence algorithm. MLAW defines a change when the measure of dissimilarity exceeds a specified limit between two windows.

Weighted Majority Algorithm (WMA) leverages a weighted vote on predictive algorithm performance from the pool [19]. Dynamic Weighted Majority (DWM) is another ensemble that complies with the rule of pure learning mistake and a user-defined factor reduces its weight [20]. When the DWM method requires, a new member of the ensemble is added to the ensemble.

Adaptive Windowing Based Online Ensemble (AWOE) [39] is proposed a hybrid approach by combining the online processing and the best components of the block-based ensemble. AWOE algorithm is applied to determine each ensemble member with an adaptive window as a sudden drift detector. In AWOE algorithm, the relative entropy (Kullback-Leibler distance) is used to detect the sudden drift by comparing the difference between two sub-windows.

III. Proposed Methodology

The main purpose of this research work is to study the adaptation of classification on the four different types of concept drift. Fig. 2 shows the architecture overview of the proposed framework. In Fig. 2, the proposed preprocessing stage has been combined with the following three steps:

1) Defining the Adaptive Windows
2) Filtering the Noisy Samples from Each Window
3) Handling the Types of Concept Drift with Ensemble Learning Based on the Adaptive Windows

Algorithm (1), (2), and (3) provide a brief description of proposed KNN embedded in each adaptive window of A-AUE2. After A-AUE2 approach has been established, a new ensemble model can be created by tackling noise and concept drifts.

1) Defining the Adaptive Windows: In this paper, the adaptive windows based on detecting sudden and gradual drifts are determined by using Brier Skill Score (BSS)

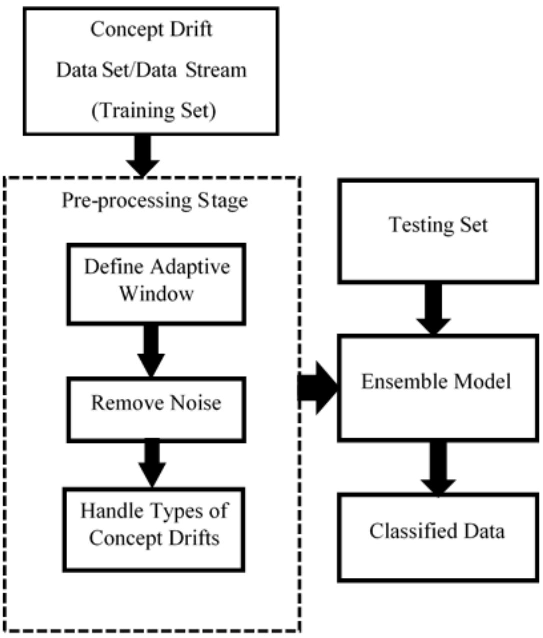


Fig. 2. Framework of Proposed Approach

method from the data stream. Brier Score (BS) and Brier Skill Score (BSS) are commonly used verification measures of forecast accuracy and skill [21], [35]. BSS is used for the skill assessment of probabilistic forecasts. BSS is based on BS, an index for the validation of probability forecasts. BSS has a range of -$\propto$ to 1. In BSS, the best value for a perfect forecast is 1. When BSS is 0, the prediction skill is equal to the reference prediction.

$$BSS = 1 - \frac{BS}{BS_{rf}} \tag{1}$$

In this Eq. (1), BS is Brier Score and BS_{rf} is Brier Score for the reference forecast. In BSS, negative values mean that the forecast is less accurate than a standard forecast. The value 0 is no skill compared to the reference forecast and the value 1 is a perfect skill compared to the reference forecast. BS measures the mean squared error between expected and forecast probability [21]-[23], [35], [46]. The score summarizes the magnitude of the error in the probability forecasts. The best possible BS is 0 for total accuracy. The lowest BS is 1, meaning that the forecast is completely incorrect.

$$BS = \frac{1}{N}\sum_{i=1}^{R}\sum_{t=1}^{N}(p_{ti} - o_{ti})^2 \tag{2}$$

In Eq. (2), *R* is the number of classes, *N* is the number of instances, p_{ti} is the forecasted probability and o_{ti} is the outcome happened 1 or 0.

In Algorithm (1), Brier Skill Score (BSS) and Brier Score (BS) methods are used to define adaptive windows based on the detecting sudden drift and gradual drift from the data stream. This paper uses the two windows (older and recent) and the recent window size is 50. The recent window (n_r) includes the probability predictions of instances. The base classifiers of BS (Brier Score) are Hoeffding Tree classifier (HT) and Hoeffding Adaptive Tree that has a

Algorithm 1 : Adaptive Windows Based on Brier Skill Score

Input: s : data stream, w : recent window, B_i : an adaptive window from data steam, BS_o : Brier Score of older window, BS_{rf} : Brier Score of recent window

Output: A new data set or An adaptive window.

storedPreds $\leftarrow$ new byte[w];
storedPreds1 $\leftarrow$ new byte[w1];
$n_o \leftarrow 0$, $n_r \leftarrow 0$, $BS_o \leftarrow 0$, $BS_{rf} \leftarrow 0$;
$w \leftarrow 50$;
changeDetected $\leftarrow$ false;
procedure ADAPTIVEWINDOW(s, B_i)
  for each $x_i \in s$ **do**
    if changeDetected **then**
      reset storedPreds, storedPreds2 ;
      $B_i \leftarrow$ instances from two windows;
      call Algorithm (2);
      changeDetected $\leftarrow$ false;
      $n_o \leftarrow 0$, $n_r \leftarrow 0$, $BS_o \leftarrow 0$, $BS_{rf} \leftarrow 0$;
    end if
    Updates probability predictions in older and recent windows;
    Updates statistics of both windows: n_o, n_r, BS_o, BS_{rf} ;
    if $n_o \geq w$ **then**
      if $BS_o \geq BS_rf$ **then**
        BSS = 1- $\frac{BS_rf}{BS_o}$;
      else
        BSS = 1- $\frac{BS_o}{BS_rf}$;
      end if
      if *BSS* $\geq$ Threshold **then**
        changeDetected $\leftarrow$ true;
      end if
    end if
  end for
end procedure

classifier at the leaves (denoted as HATCL). If the recent window is full, the probability predictions of instances slide to insert to the older window (n_o). n_o includes the overall probability predictions of instances except the recent window of probability predictions of instances. If the recent window and the older window are 50, the BS of each window is calculated by using the probability prediction of the classifier. BS_{rf} is the BS for the recent window and BS_o is the BS for the older window. After Brier Scores of the two windows are calculated, the Brier Skill Score (BSS) is calculated. The percentage of BSS is matched with the predefined threshold to define whether or not drift. If the percentage of BSS is greater than or equal predefined threshold, a drift (sudden or gradual drift) has occurred.

When the drift is detected, the instances are combining from the two windows to define an adaptive window or chunk (B_i). When the drift is not encountered, each of the two windows slides instance by instance from the data stream. But, if a sudden or gradual drift is not detected in 1000 instances, 1000 instances are used to recognize a window from the data stream. After defining each window, Algorithm (2) is called to filter noise from this window.

Algorithm (1) reduces the difficultly of determining the appropriate windows for the chunk-based ensemble. This

Algorithm 2 : Handling the Noisy Samples over Each Window Based on KNN

Input: k : number of nearest neighbors, n_k : number of nearest neighbors instances, B_i : instances of a window from Algorithm (1), B_i' : new chunk.
Output: A new data set (a new window) after filtering noise.

```
k ← 6;
procedure NEWDATASET(B_i, B'_i)
    for each x_j ∈ B_i do
        n_k(x_j) = KNN(x_j, B_i, k);
        for each y_j ∈ n_k do
            if class(x_j) = class(y_j) then
                count = count+1;
            end if
        end for
        if count > 1 then
            B'_i = x_j;
        end if
    end for
    call Algorithm (3);
end procedure
```

algorithm incorporates to aid the handling of the four types of concept drift based on ensemble learning.

2) Filtering the Noisy Samples from Each Window: K-Nearest Neighbors (KNN) [47]-[48] is used to filter the noisy samples from data streams. Noisy samples in real environments are inevitable because of inaccuracy in data collection, device weaknesses, data transmission and man-made disruptions. Therefore, concept drifts and noisy samples are the two majors in the challenges of data streams. The problem of noisy sample handling in concept-drifting data streams has been increasingly concerned. Noisy sample processing is one of the most important fields in mining data streams. Noise can be interpreted as fluctuations in time or space that confuse drift detectors. A difficult problem in handling the concept drifts is to distinguish between true concept drift and noise.

In Algorithm (2), KNN filters any samples whose class label is equal one or below from the class of its six nearest neighbors (k = 6) over each window from Algorithm (1). After defining a new window, Algorithm (3) is used. This algorithm is a more effective approach for ensemble learning by combining in an adaptive window.

3) Handling the Types of Concept Drift with Ensemble Learning Based on the Adaptive Windows: Different four types of concept drift are handled by using ensemble learning dependent on the adaptive windows. Accuracy Updated Ensemble-2 (AUE2) depends on the fixed chunk-based ensemble approach. After creating a new classifier from the incoming new window and calculating weight from each fixed-size data chunk, this classifier is added to the ensemble. If the number of component classifiers in the ensemble is full, the poorest classifier is substituted with the new classifier. After the weakest classifier has been substituted with a new classifier, the weights of the remaining classifiers in the ensemble are adjusted by using the instances of the incoming window. By using the prediction errors of the incoming window, the weights of each component classifier in the ensemble are evaluated. The remaining classifiers of the ensemble are incrementally trained with the new incoming window. AUE2 retains a weighted component classifier pool and predicts the incoming instance class by incorporating the component predictions with a weighted voting rule.

$$MSE_r = \sum p(y)(1 - p(y))^2 \tag{3}$$

$$MSE_{ij} = \frac{1}{|B_i'|} \sum_{x,y \in B_i'} (1 - f_y^j(x))^2 \tag{4}$$

$$w_{c'} = \frac{1}{MSE_r + \epsilon} \tag{5}$$

$$w_{ij} = \frac{1}{MSE_r + MSE_{ij} + \epsilon} \tag{6}$$

In this paper, let B_1', B_2', . . . , B_i' be an adaptive window that contains instances from Algorithm (2). MSE_r from Eq. (3) is the mean square error of the current class distribution from current the adaptive window from Algorithm (2). MSE_{ij} from Eq. (4) is the prediction error of classifier from the ensemble by using the adaptive incoming window B_i'. Function ${f_y}^j$(x) from Eq. (4) denotes the probability given by component classifier $C_j \in \varepsilon$ (j = 1, 2, . . . , k) from ensemble that *x* is an instance of class *y*. $w_{c'}$ from Eq. (5) is the weight of classifier of new window from Algorithm (2) and w_{ij} from Eq. (6) is the weight of component classifiers from the ensemble by using the incoming window. In addition, a very small positive value (ϵ) is added to the equation in order to avoid the division by zero problems.

In Algorithm (3), C' is created the classifier from the instances of the incoming window. Then, new weight $w_{c'}$ is calculated with the incoming window B_i' and C' is added to the ensemble. If the ensemble is full, the least accuracy is substituted with C' in the ensemble. After substituting, the remaining component classifiers $C_j \in \varepsilon$ in the ensemble are incrementally trained with the incoming window B_i'. In this algorithm, the ensemble size (*en*) is used 10. If the memory boundary is exceeded, the least active leaves from Hoeffding are pruned to match the memory restriction.

In Algorithm (3), A-AUE2 approach based on the adaptive windows is proposed by creating the new learners and calculating the weights of learners from the incoming windows of Algorithm (2). This Algorithm (3) is a more effective approach for handling the four types of concept drift because this paper uses A-AUE2 (AUE2 approach based on the adaptive windows).

IV. EXPERIMENTAL SETUP

This section presents the related information for the analysis of experiments. All of the tested algorithms are implemented in Java programming language by extending the MOA software [24]. All evaluation measures are determined periodically using the prequential evaluation method with the basic classification performance evaluator. The ensemble size for AWE, Anticipative Dynamic Adaptation to Concept Change (ADACC), Adaptable Diversity-based Online Boosting (ADOB), Dynamic Adaptation to Concept Changes (DACC) and AUE2 is 10. VFDT or HT and HATCL are used for the base classifiers of all the single and ensemble drift detection and handling methods. The experiment analysis

Algorithm 3 A-AUE2 (AUE2 Based on Adaptive Windows)

Input: *en* : number of ensemble members, *m* : memory limit, B_i' : instances of the incoming window from Algorithm (2).
Output: ε : ensemble of *en* weighted incremental classifiers.

$\varepsilon \leftarrow \phi$;
$C' \leftarrow$ new component classifier built on B_i' ;
procedure ENSEMBLECLASSIFIERS(B_i', ε)
    calculate $w_{c'}$ based on Eq. (5);
    for each $C_j \in \varepsilon$ **do**
        apply C_j on B_i' to derive MSE_{ij} ;
        compute weight w_{ij} based on Eq. (6);
    end for
    if $\mid \varepsilon \mid < en$ **then**
        $\varepsilon \leftarrow \varepsilon \cup C'$;
    else
        substitute least accurate classifier in ε with C' ;
    end if
    for each $C_j \in \varepsilon \setminus C'$ **do**
        incrementally train classifier C_j with B_i';
    end for
    if memory-usage $(\varepsilon) > m$ **then**
        prune (decrease size of) component classifiers;
    end if
end procedure

uses the default parameters for the base classifier of single and ensemble drift detection approaches in the MOA tool. The window size of all ensemble algorithms is 1000. In this paper, the predefined threshold (72) is used to determine a drift by comparing two windows (recent and older).

A. Data Sets

This paper uses 10 concept drift data sets. Table I provides the main characteristics of these data sets. This paper utilizes two kinds of data sets.

There are:

1) Real Data Sets or Data Streams
2) Synthesize Data Sets or Data Streams

1) Real Data Sets or Data Streams: The real benchmarks data sets are gathered from the UCI Repository [1]. The first real data set is Usenet data set related to medicine, baseball, and space domains. Usenet data set contains 1500 instances, 99 attributes, and two classes.

The second real benchmark data set is the Shuttle data set and it contains 9 attributes all of which are numerical. This data set consists of 3 classes and 100000 instances.

The third real data set is Weather data set and it contains eight different features such as temperature, pressure and wind speed, etc. from 1949 to 1999 were measured at the Offutt Air Force Base in Bellevue, Nebraska. The aim of Weather data set is to forecast whether or not it will rain on a certain day. This data set has 18159 instances.

The final real data set is Airlines [10], [27] and it is a binary data set consisting of 539,383 samples. The aim is to estimate whether flights are delayed or not, based on a set of flight information: name of the company, departure time, flight number, duration, airports of origin and destination and day of the week.

2) Synthesize Data Sets or Data Streams: The synthesis data sets are generated using MOA framework.

The SEA generator is used to create three data sets. These three data sets are SEA_S with sudden concept drifts, SEA_{SR} with sudden recurrent drifts, and SEA_G with gradual drifts. The three data sets contain 1000000 instances, 10% noise, and 2 classes. The three data sets consist of three attributes and the first two attributes are relevant. SEA_S data set has three concept drift and SEA_{SR} data set has four concept drifts with a sudden drift every 250000 instances. SEA_G produces 9 concept drifts.

The random radius basis function generator (RanRBF) creates a fixed number of random centroids, class labels, position, weight, and standard deviation. We use this generator to generate gradual drift ($RanRBF_{GR}$) which has 1000000 instances, 20 features and no noise. $RanRBF_{GR}$ is designed to contain four gradual recurring drifts with each concept containing four decision classes.

Hyperplane (Hyper) is mainly used to simulate incremental drift. We use this generator with 1000000 samples and 10 attributes. The incremental drift can be generated by changing weight by 0.1 for each sample and by adding 5% noise to the data.

This paper uses the Random Tree generator to create $RanTree_{SRF}$ with 10 numerical attributes, 6 classes, and no noise. The $RanTree_{SRF}$ data set comprises only 100000 samples, but 15 sudden drifts with a sudden drift every 2790 observations, it is the fastest evolving data set.

B. Compared Methods

The proposed approach can be compared to the other algorithms such as VFDT or HT, HATCL, NB, Adaptive Sliding Window (ADWIN), ADOB, ADACC, AWE, DACC and AUE2 in many experiments. ADWIN is concept drifts detecting approach based on a single classifier. ADOB, ADACC, AWE, DACC and AUE2 are ensemble models to be used for adapting the types of concept drift.

Naive Bayes (NB) [26] is a simple classifier without ignoring mechanisms that are useful for handling stationary data streams. VFDT or HT [25] is a well-established tree decision model which is specialized in high-speed data streams. HATCL can be used to enhance prediction accuracy when they are applied feature drifting data streams [40]. ADWIN approach [3] observes the changes and maintains a data stream with the modified statistics. ADOB [41] approach designed to resolve frequent and sudden concept drifts more effectively in online learning environments. ADACC method [32] may use for the recurring concepts and build on an ensemble-based adaptive online learner. DACC approach [42] tackles the difficulty of finding the appropriate threshold and the concept drifts with many levels of severity and speed.

C. Performance Metrics

Precision from Eq. (7) is the percentage of the expected drifts that occur. Recall from Eq. (8) is the percentage of drifts that each method has correctly detected. F1 from Eq. (9) is the harmonic mean of Precision and Recall. The

[1] htt, http://rchive.ics.uci.edu/ml

TABLE I
CHARACTERISTICS OF EXPERIMENTAL DATA SETS

Data Sets	# Inst	# Att	# Classes	Noise	# Drift	Drift Type
SEA_G	1000000	3	2	10%	9	gradual
SEA_S	1000000	3	2	10%	3	sudden
$Hyper_I$	1000000	10	2	5%	1	incremental
SEA_{SR}	1000000	3	2	10%	4	sudden recurrent
$RanRBF_{GR}$	1000000	20	4	0%	4	gradual recurrent
$RanTree_{SRF}$	100000	10	6	0%	15	sudden recurrent
$Usenet2$	1500	99	2	-	-	-
$Shuttle$	100000	9	3	-	-	-
$Weather$	18159	8	2	-	-	-
$Airlines$	539,383	7	2	-	-	-

TABLE II
AVERAGE ACCURACY RESULTS FROM 10 DATA SETS USING DRIFT DETECTION AND HANDLING METHODS BASED ON HT

Data Sets	HT	NB	ADWIN	ADOB	ADACC	AWE	DACC	AUE2	A-AUE2
SEA_G	85.68	84.71	48.73	50.26	83.15	86.09	83.12	88.40	**88.91**
SEA_S	84.89	83.87	51.88	56.74	84.20	85.64	84.17	88.94	**89.29**
$Hyper_I$	87.53	78.79	50.04	50.45	87.02	92.19	86.81	93.12	**93.14**
SEA_{SR}	87.25	86.56	42.19	47.29	84.22	85.48	84.26	88.97	**89.34**
$RanRBF_{GR}$	91.52	60.39	24.01	29.69	62.76	78.41	62.70	**97.24**	96.61
$RanTree_{SRF}$	46.28	43.41	28.36	28.39	**57.18**	54.36	**57.18**	51.19	51.35
$Usenet2$	72.00	72.13	72.00	64.87	**76.33**	67.07	76.27	67.07	67.47
$Shuttle$	62.58	44.44	34.94	44.78	94.80	93.42	94.48	95.03	**95.22**
$Weather$	73.43	69.22	68.80	72.54	73.28	69.81	73.32	74.19	**74.59**
$Airlines$	63.89	58.48	55.45	54.35	59.99	60.66	60.01	**64.51**	63.96

TABLE III
AVERAGE ACCURACY RESULTS FROM 10 DATA SETS USING DRIFT DETECTION AND HANDLING METHODS BASED ON HATCL

Data Sets	HATCL	NB	ADWIN	ADOB	ADACC	AWE	DACC	AUE2	A-AUE2
SEA_G	85.98	84.71	52.49	58.98	84.67	84.62	84.72	86.53	**86.95**
SEA_S	86.66	83.87	51.37	76.54	83.91	83.78	83.83	87.66	**87.67**
$Hyper_I$	88.22	78.79	50.02	50.47	80.78	78.81	79.12	92.03	**92.06**
SEA_{SR}	86.28	86.56	42.40	51.58	86.56	86.49	86.56	87.63	**87.79**
$RanRBF_{GR}$	84.51	60.39	28.69	29.68	62.11	59.20	60.49	90.16	**90.33**
$RanTree_{SRF}$	42.91	43.41	28.34	28.39	**49.29**	43.72	43.43	46.01	46.33
$Usenet2$	72.53	72.13	43.93	72.33	**73.40**	70.00	72.53	70.00	66.47
$Shuttle$	46.75	44.44	35.05	44.29	55.47	45.49	54.34	80.37	**87.48**
$Weather$	71.10	69.22	68.70	**72.31**	71.36	68.35	69.46	68.80	70.66
$Airlines$	58.36	58.48	55.45	54.05	59.14	58.42	58.77	**62.15**	61.95

imbalance ratio between the numbers of positive and negative samples is extremely impact of Matthews Correlation Coefficient (MCC) criteria from Eq. (10) [43]-[45]. The value in the [-1, 1] interval is returned for MCC and is based upon the values True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN)) in the confusion matrix. Area Under the ROC Curve (AUC) Eq. (11) [22], [49]-[50] represents a trade-off measure between TP rates and FP rates. It estimates the area under the receiver operating characteristics (ROC) curves. ROC is obtained by plotting a set of TP rates versus FP rates related to various classification thresholds.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{9}$$

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \tag{10}$$

$$AUC = \frac{1 + TPrate - FPrate}{2} \tag{11}$$

Additionally, Wilcoxon Signed Rank Test and Friedman Test are used for comparing A-AUE2 and the comparative methods. Wilcoxon Signed Rank Test is a hypothetical test of non-parametric statistics that used to compare two related samples from the same population and assessed for significant differences (i.e. It is a paired difference test) [22], [29], [31], [34]. Wilcoxon Signed Rank Test is also used with SPSS software to help the statistical analysis

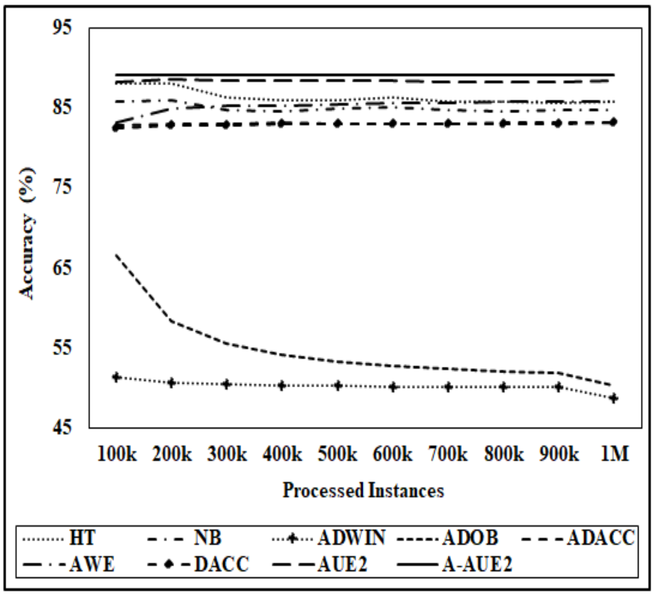


Fig. 3. Classification accuracy of drift detection and handling methods using HT as base classifier for SEA_G data set

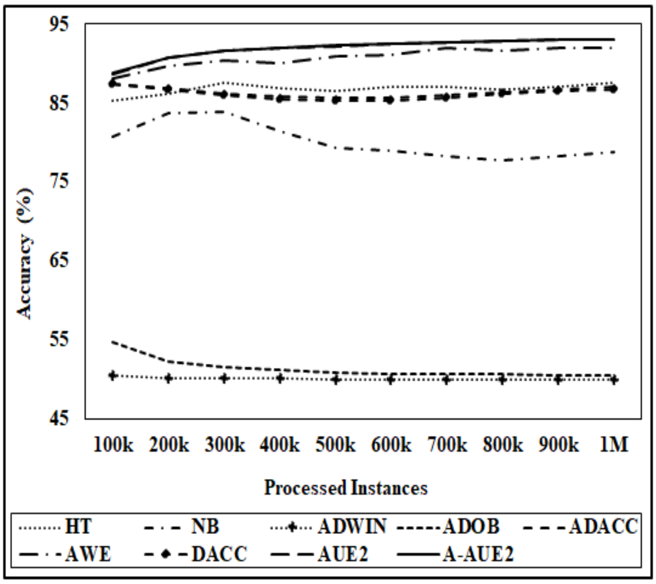


Fig. 5. Classification accuracy of detection and handling methods using HT as base classifier for $Hyper_I$ data set

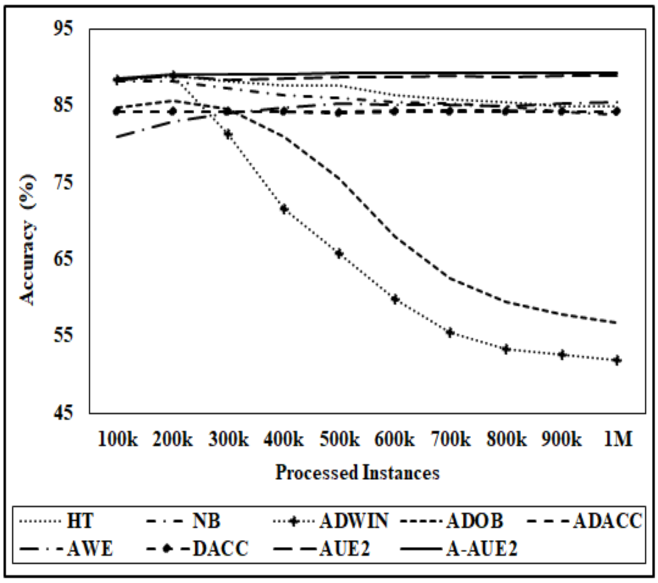


Fig. 4. Classification accuracy drift detection and handling methods using HT as base classifier for SEA_S data set

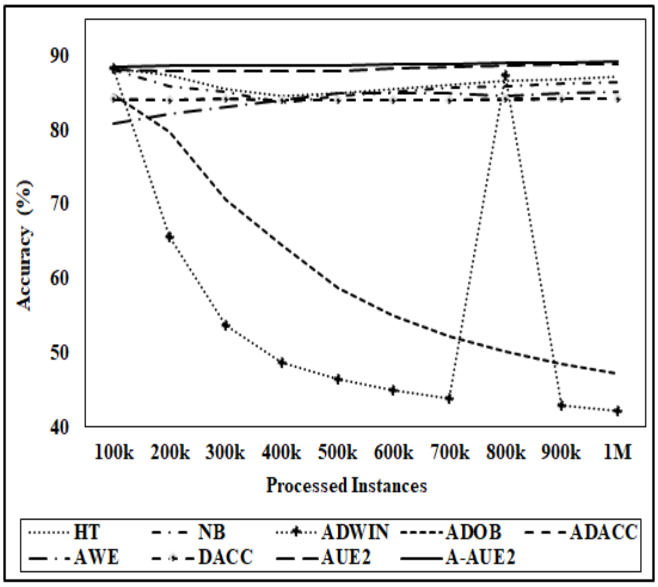


Fig. 6. Classification accuracy of drift detection and handling methods using HT as base classifier for SEA_{SR} data set

of comparative results. Friedman Test is evaluated on the statistical significance of the differences in the algorithms accuracies [11], [18], [29].

V. Results and Discussions

This section illustrates the results analysis arrangements for evaluating A-AUE2 according to the results of performance analysis and statistical analysis using two base learners: HT and HATCL. The best case for each data set is highlighted in bold.

A. *Performance Analysis*

Table II and III illustrate the average prediction results for the different approaches using HT and HATCL, respectively, as base learner. Additionally, the two graphical plots are generated for each data set to show the performance curves of all tested algorithms. Figs. 3-8 report the best performance of A-AUE2 in Table II. Figs. 9-14 indicate the highest performance of A-AUE2 in Table III. In Figs. 3-8 and Figs. 9-14, the x-axis represents the number of observes processed and the y-axis depicts the Average accuracy. Figs. 3-14 show the prequential Average accuracy of the tested algorithms on data sets with the four types of concept drift.

According to Table II, Fig. 3 indicates the performance of the algorithms in the SEA_G data set. The very worst-performing algorithm for gradually drifted data streams is ADWIN, followed by ADOB, ADACC and DACC. Fig. 4 illustrates the Average accuracy of the SEA_S data set, which involves three sudden concept drifts. In Fig. 4, the most severely malfunctioning algorithm is ADWIN, followed by

TABLE IV
EVALUATION RESULTS OF DRIFT DETECTION AND HANDLING METHODS USING HT AND HATCL AS BASE CLASSIFIER

DataSets	Hoeffding Tree(HT)						Hoeffding Adaptive Tree Classifier Leaves (HATCL)					
	Methods	**Precision**	**Recall**	**F1**	**MCC**	**AUC**	**Methods**	**Precision**	**Recall**	**F1**	**MCC**	**AUC**
SEA_G	HT	0.85704	0.85725	0.85715	0.71429	0.85725	HATCL	0.85965	0.85988	0.85976	0.71953	0.85988
	NB	0.84706	0.84733	0.84719	0.69438	0.84733	NB	0.84706	0.84733	0.84719	0.69438	0.84733
	ADWIN	0.67658	0.50124	0.57586	0.02957	0.50124	ADWIN	0.69107	0.53742	0.60464	0.16912	0.53742
	ADOB	0.55884	0.51470	0.53586	0.05882	0.51470	ADOB	0.63712	0.59744	0.61664	0.23118	0.59744
	ADACC	0.83144	0.83169	0.83156	0.66312	0.83169	ADACC	0.84659	0.84683	0.84671	0.69342	0.84683
	AWE	0.86076	0.86096	0.86086	0.72173	0.86096	AWE	0.84623	0.84650	0.84637	0.69273	0.84650
	DACC	0.83119	0.83144	0.83131	0.66262	0.83144	DACC	0.84716	0.84743	0.84730	0.69459	0.84743
	AUE2	0.88388	0.88398	0.88393	0.76786	0.88398	AUE2	0.86529	0.86557	0.86543	0.73087	0.86557
	A-AUE2	**0.88900**	**0.88906**	**0.88903**	**0.77806**	**0.88906**	A-AUE2	**0.86936**	**0.86963**	**0.86949**	**0.73899**	**0.86963**
SEA_S	HT	0.84687	0.83067	0.83869	0.67734	0.83067	HATCL	0.86600	0.84983	0.85784	0.71565	0.84983
	NB	0.84228	0.81367	0.82773	0.65533	0.81367	NB	0.84228	0.81367	0.82773	0.65533	0.81367
	ADWIN	0.66700	0.59944	0.63142	0.25774	0.59944	ADWIN	0.66024	0.59454	0.62567	0.24616	0.59454
	ADOB	0.61911	0.61156	0.61531	0.23054	0.61156	ADOB	0.75906	0.77220	0.76557	0.53110	0.77220
	ADACC	0.83953	0.82301	0.83119	0.66233	0.82301	ADACC	0.84506	0.81233	0.82837	0.65658	0.81233
	AWE	0.85648	0.83734	0.84680	0.69356	0.83734	AWE	0.84113	0.81273	0.82669	0.65324	0.81273
	DACC	0.83884	0.82310	0.83089	0.66175	0.82310	DACC	0.84185	0.81318	0.82727	0.65440	0.81318
	AUE2	0.88992	0.87501	0.88240	0.76478	0.87501	AUE2	**0.87600**	0.86142	0.86865	0.73728	0.86142
	A-AUE2	**0.89305**	**0.87931**	**0.88613**	**0.77224**	**0.87931**	A-AUE2	0.87595	**0.86160**	**0.86872**	**0.73741**	**0.86160**
$Hyper_I$	HT	0.87534	0.87534	0.87534	0.75068	0.87534	HATCL	0.88306	0.88217	0.88262	0.76523	0.88217
	NB	0.78791	0.78791	0.78791	0.57582	0.78791	NB	0.78791	0.78791	0.78791	0.57582	0.78791
	ADWIN	0.62120	0.50041	0.55430	0.01403	0.50041	ADWIN	0.60394	0.50026	0.54723	0.01032	0.50026
	ADOB	0.50458	0.50458	0.54335	0.04028	0.50458	ADOB	0.64373	0.50469	0.56579	0.05192	0.50469
	ADACC	0.87025	0.87022	0.87023	0.74046	0.87022	ADACC	0.80776	0.80775	0.80776	0.61551	0.80775
	AWE	0.92190	0.92190	0.92190	0.84380	0.92190	AWE	0.78813	0.78813	0.78813	0.57626	0.78813
	DACC	0.86817	0.86812	0.86815	0.73629	0.86812	DACC	0.79118	0.79118	0.79118	0.58236	0.79118
	AUE2	0.93121	0.93121	0.93121	0.86242	0.93121	AUE2	0.92027	0.92027	0.92027	0.84053	0.92027
	A-AUE2	**0.93139**	**0.93139**	**0.93139**	**0.86278**	**0.93139**	A-AUE2	**0.92064**	**0.92063**	**0.92064**	**0.84127**	**0.92063**
SEA_{SR}	HT	0.86870	0.85441	0.86150	0.72297	0.85441	HATCL	0.86014	0.84126	0.85060	0.70115	0.84126
	NB	0.86559	0.84211	0.85369	0.70732	0.84211	NB	0.86559	0.84211	0.85369	0.70732	0.84211
	ADWIN	0.63681	0.53767	0.58305	0.14357	0.53767	ADWIN	0.62749	0.53813	0.57938	0.13944	0.53813
	ADOB	0.58995	0.55892	0.57402	0.14560	0.55892	ADOB	0.62938	0.59565	0.61205	0.22249	0.59565
	ADACC	0.83777	0.81795	0.82774	0.65542	0.81795	ADACC	0.86643	0.84127	0.85366	0.70725	0.84127
	AWE	0.85645	0.82748	0.84171	0.68331	0.82748	AWE	0.86464	0.84135	0.85284	0.70561	0.84135
	DACC	0.83862	0.81796	0.82816	0.65626	0.81796	DACC	0.86564	0.84202	0.85367	0.70727	0.84202
	AUE2	0.88874	0.87207	0.88033	0.76063	0.87207	AUE2	0.87413	0.85711	0.86554	0.73105	0.85711
	A-AUE2	**0.89324**	**0.87571**	**0.88439**	**0.76875**	**0.87571**	A-AUE2	**0.87701**	**0.85784**	**0.86732**	**0.73460**	**0.85784**
$RanRBF_{GR}$	HT	0.91464	0.91526	0.91495	0.88656	0.94346	HATCL	0.84537	0.84192	0.84364	0.79181	0.89505
	NB	0.60065	0.59616	0.59840	0.4660	0.73199	NB	0.60065	0.59616	0.59840	0.46601	0.73199
	ADWIN	0.60283	0.25511	0.35851	0.04500	0.50340	ADWIN	0.69628	0.30136	0.42065	0.18095	0.53411
	ADOB	0.75172	0.31156	0.44053	0.21064	0.54088	ADOB	0.75810	0.31144	0.44150	0.21176	0.54081
	ADACC	0.63770	0.62553	0.63156	0.50569	0.74985	ADACC	0.61940	0.61290	0.61613	0.48939	0.74324
	AWE	0.78654	0.78461	0.78558	0.71298	0.85595	AWE	0.59220	0.58463	0.58839	0.45190	0.72428
	DACC	0.63714	0.62496	0.63099	0.50493	0.74947	DACC	0.60268	0.59724	0.59995	0.46786	0.73272
	AUE2	**0.97285**	**0.97205**	**0.97245**	**0.96319**	**0.98137**	AUE2	0.90225	0.90058	0.90142	0.86843	0.93376
	A-AUE2	0.96645	0.96556	0.96600	0.95463	0.97707	A-AUE2	**0.90419**	**0.90206**	**0.90312**	**0.87071**	**0.93478**
$RanTree_{SRF}$	HT	0.40645	0.32416	0.36067	0.24846	0.60341	HATCL	0.37044	0.28239	0.32048	0.20358	0.57888
	NB	0.38559	0.28159	0.32548	0.20624	0.57834	NB	0.38558	0.28159	0.32548	**0.32548**	0.57834
	ADWIN	**0.58951**	0.17688	0.27211	0.06204	0.50619	ADWIN	0.55375	0.17647	0.26765	0.05836	0.50596
	ADOB	0.55579	0.17751	0.26908	0.06320	0.50655	ADOB	**0.56125**	0.17725	0.26941	0.06342	0.50642
	ADACC	0.50621	0.41892	0.45845	0.36950	**0.66297**	ADACC	0.48127	**0.37001**	**0.41837**	0.32362	**0.62849**
	AWE	0.50378	**0.42283**	**0.45977**	**0.36998**	0.66211	AWE	0.38885	0.28819	0.33104	0.21755	0.58167
	DACC	0.50618	0.41889	0.45841	0.36948	0.66295	DACC	0.38355	0.28190	0.32496	0.20624	0.57855
	AUE2	0.46104	0.37501	0.41360	0.3160	0.63436	AUE2	0.42328	0.32252	0.36609	0.25360	0.60257
	A-AUE2	0.48999	0.35614	0.41248	0.30144	0.62472	A-AUE2	0.43398	0.31841	0.36732	0.24885	0.60083

	HT	0.70231	0.61650	0.65661	0.30705	0.61650	HATCL	0.69159	0.65050	0.67042	0.33962	0.65050
	NB	0.68546	0.64750	0.66594	0.33079	0.64750	NB	0.68546	0.64750	0.66594	0.33079	0.64750
	ADWIN	0.70231	0.61650	0.65661	0.30705	0.61650	ADWIN	0.38048	0.37700	0.37873	-0.24250	0.37700
	ADOB	0.62469	0.63650	0.63054	0.26092	0.63650	ADOB	0.70710	**0.73000**	**0.71837**	**0.43650**	**0.73000**
$Usenet2$	ADACC	**0.73694**	**0.70950**	**0.72296**	**0.44560**	**0.70950**	ADACC	0.70289	0.66300	0.68236	0.36371	0.66300
	AWE	0.69227	0.50800	0.58599	0.07844	0.50800	AWE	**0.76990**	0.55550	0.64536	0.24478	0.55550
	DACC	0.73601	0.70900	0.72225	0.44419	0.70900	DACC	0.69068	0.65350	0.67158	0.34217	0.65350
	AUE2	0.69227	0.50800	0.58599	0.07844	0.50800	AUE2	**0.76990**	0.55550	0.64536	0.24478	0.55550
	A-AUE2	0.63740	0.52450	0.57547	0.11604	0.52450	A-AUE2	0.57963	0.52000	0.54820	0.07982	0.52000
	HT	0.62728	0.62235	0.62480	0.43822	0.71710	HATCL	0.48381	0.47226	0.47797	0.21608	0.60410
	NB	0.45731	0.44173	0.44938	0.16881	0.58063	NB	0.45731	0.44173	0.44938	0.16881	0.58063
	ADWIN	0.75636	0.36522	0.49259	0.15560	0.52398	ADWIN	0.75630	0.36637	0.49361	0.15844	0.52484
	ADOB	0.78395	0.46061	0.58027	0.32934	0.59569	ADOB	0.77795	0.45589	0.57489	0.32087	0.59213
$Shuttle$	ADACC	0.94797	0.94843	0.94820	0.92218	0.96123	ADACC	0.54840	0.55017	0.54928	0.32950	0.66289
	AWE	0.93400	0.93418	0.93409	0.90118	0.95065	AWE	0.47815	0.44773	0.46244	0.18829	0.58586
	DACC	0.94475	0.94508	0.94492	0.91724	0.95872	DACC	0.54229	0.54088	0.54158	0.31464	0.65565
	AUE2	0.95016	0.95050	0.95033	0.92549	0.96286	AUE2	0.80622	0.80516	0.80569	0.70757	0.85354
	A-AUE2	**0.95202**	**0.95227**	**0.95214**	**0.92821**	**0.96419**	A-AUE2	**0.87566**	**0.87583**	**0.87575**	**0.81319**	**0.90672**
	HT	0.68840	0.66123	0.67454	0.34857	0.66123	HATCL	0.65924	0.64664	0.65288	0.30563	0.64664
	NB	0.65206	0.66277	0.65738	0.31466	0.66277	NB	0.65206	0.66277	0.65738	0.31466	0.66277
	ADWIN	0.61032	0.51666	0.55960	0.08574	0.51666	ADWIN	0.60219	0.51604	0.55580	0.08097	0.51604
	ADOB	**0.70504**	**0.73422**	**0.71933**	**0.43829**	**0.73422**	ADOB	**0.68197**	**0.68939**	**0.68566**	**0.37129**	**0.68939**
$Weather$	ADACC	0.68715	0.67563	0.68134	0.36260	0.67563	ADACC	0.67243	0.68096	0.67667	0.35329	0.68096
	AWE	0.63923	0.55172	0.59226	0.16972	0.55172	AWE	0.64020	0.64805	0.64410	0.28814	0.64805
	DACC	0.68774	0.67647	0.68206	0.36404	0.67647	DACC	0.65587	0.66798	0.66187	0.32362	0.66798
	AUE2	0.70497	0.64763	0.67508	0.34790	0.64763	AUE2	0.65556	0.67254	0.66394	0.32767	0.67254
	A-AUE2	0.70400	0.67466	0.68902	0.37752	0.67466	A-AUE2	0.66692	0.67787	0.67235	0.34461	0.67787
	HT	0.63967	0.61818	0.62874	0.25695	0.61818	HATCL	0.59017	0.54560	0.56701	0.12825	0.54560
	NB	0.59109	0.54736	0.56839	0.13136	0.54736	NB	0.59109	0.54736	0.56839	0.13136	0.54736
	ADWIN	0.46959	0.49999	0.48431	-0.00085	0.49999	ADWIN	0.51041	0.50001	0.50516	0.00073	0.50001
	ADOB	0.47704	0.49503	0.48587	-0.02136	0.49503	ADOB	0.46902	0.49267	0.48055	-0.03014	0.49267
$Airlines$	ADACC	0.59202	0.58741	0.58971	0.17937	0.58741	ADACC	0.59684	0.55661	0.57603	0.14809	0.55661
	AWE	0.59911	0.59394	0.59651	0.19298	0.59394	AWE	0.59028	0.54655	0.56757	0.12965	0.54655
	DACC	0.59227	0.58766	0.58995	0.17987	0.58766	DACC	0.59465	0.55106	0.57203	0.13904	0.55106
	AUE2	**0.64352**	**0.62765**	**0.63549**	0.27070	**0.62765**	AUE2	**0.61552**	**0.60660**	**0.61103**	**0.22194**	**0.60660**
	A-AUE2	0.63635	0.62317	0.62969	**0.62969**	0.62317	A-AUE2	0.61350	0.60377	0.60856	0.21698	0.60377

TABLE V
AVERAGE ACCURACY RANKS USING HT AS BASE CLASSIIFER IN THE FRIEDMAN TEST

Data Sets	**HT**	**NB**	**ADWIN**	**ADOB**	**ADACC**	**AWE**	**DACC**	**AUE2**	**A-AUE2**
SEA_G	4	5	9	8	6	3	7	2	**1**
SEA_S	4	7	9	8	5	3	6	2	**1**
$Hyper_I$	4	7	9	8	5	3	6	2	**1**
SEA_{SR}	3	4	9	8	7	5	6	2	**1**
$RanRBF_{GR}$	3	7	9	8	5	4	6	**1**	2
$RanTree_{SRF}$	6	7	9	8	**1.5**	3	**1.5**	5	4
$Usenet2$	4.5	3	4.5	9	**1**	7.5	2	7.5	6
$Shuttle$	6	8	9	7	3	5	4	2	**1**
$Weather$	3	8	9	6	5	7	4	2	**1**
$Airlines$	3	7	8	9	6	4	5	**1**	2
Average Rank	4.1	6.3	8.5	7.9	4.5	4.5	4.8	2.7	**2.0**

ADOB. The subsequent declines in the accuracy of ADWIN and ADOB algorithms suggest that classifiers with sudden drifts do not learn from the data without any drift reaction mechanism. Additionally, AUE2 is developed with sudden changes to perform the second-best in this figure.

Fig. 5 reports on the $Hyper_I$ data set of the accuracy of the analyzed algorithms that involves incremental concept drifts. There is no drift reaction mechanism for ADWIN and ADOB classifiers resulting in the worst performance on the $Hyper_I$ data set.

Fig. 6 illustrates the analyzed prequential accuracy which includes four sudden recurring concept drifts. In this figure, ADWIN and ADOB cannot respond to sudden recurrent changes immediately and perform the worst.

TABLE VI
AVERAGE RANKS OF AVERAGE ACCURACY, PRECISION, RECALL, F1, MCC AND AUC USING HT AS BASE CLASSIFIER IN THE FRIEDMAN TEST

	HT	NB	ADWIN	ADOB	ADACC	AWE	DACC	AUE2	A-AUE2
Average accuracy	4.1	6.3	8.5	7.9	4.5	4.5	4.8	2.7	**2.0**
Precision	4.4	6.9	7.0	6.7	4.9	4.6	5.3	**2.7**	**2.7**
Recall	4.5	6.0	8.6	6.9	4.4	4.5	4.6	3.2	**2.5**
F1	4.4	6.3	8.0	7.5	4.4	4.4	4.8	2.9	**2.5**
MCC	4.3	6.2	8.4	7.2	4.4	4.5	4.8	3.2	**2.2**
AUC	4.5	6.0	8.6	6.9	4.3	4.7	4.5	3.2	**2.5**

TABLE VII
AVERAGE RANKS OF AVERAGE ACCURACY, PRECISION, RECALL, F1, MCC AND AUC USING HATCL AS BASE CLASSIFIER IN THE FRIEDMAN TEST

	HATCL	NB	ADWIN	ADOB	ADACC	AWE	DACC	AUE2	A-AUE2
Average accuracy	4.4	5.6	8.8	7.0	3.2	6.5	4.5	3.0	**2.2**
Precision	5.1	6.3	7.2	5.4	4.0	6.5	5.5	2.5	**2.4**
Recall	4.8	5.5	8.8	6.5	3.8	6.2	4.5	2.6	**2.2**
F1	5.1	5.6	8.5	6.2	3.5	6.6	4.7	2.7	**2.1**
MCC	5.0	5.1	8.9	6.3	3.5	6.6	4.7	2.8	**2.3**
AUC	4.8	5.6	8.9	6.5	3.8	6.2	4.5	2.6	**2.2**

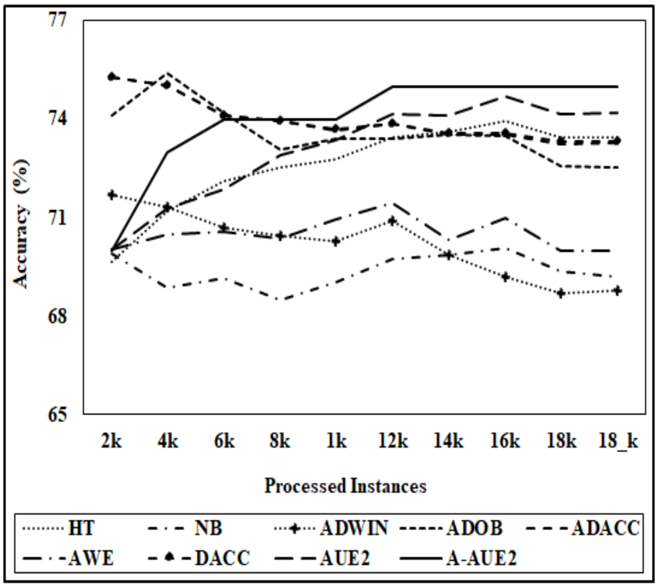


Fig. 7. Classification accuracy of drift detection and handling methods using HT as base classifier for Weather data set

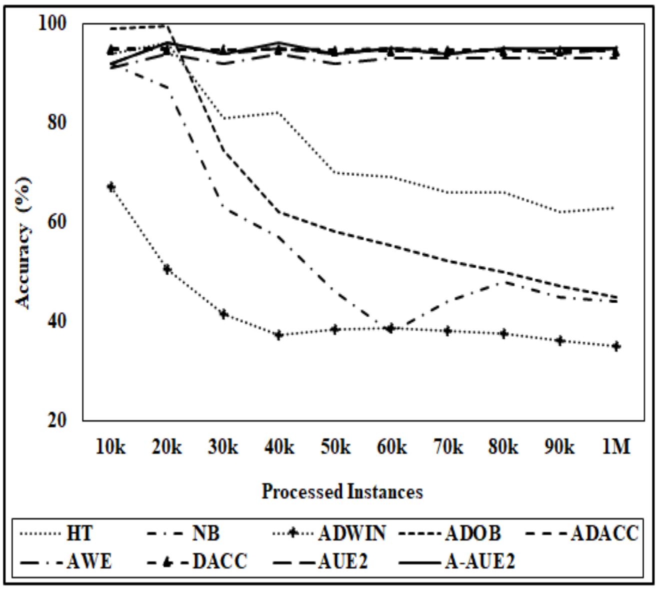


Fig. 8. Classification accuracy of drift detection and handling methods using HT as base classifier for Shuttle data set

Fig. 7 displays the performance of the evaluated algorithms for Weather data set. NB, ADWIN and AWE are the poorest performance on Weather data set. Fig. 8 shows the Average accuracy of the algorithms evaluated on Shuttle data set. NB, ADWIN, ADOB and HT are the worst-performing algorithms in this figure.

According to the detailed results from Table III using HATCL base classifier, Fig. 9 reports the accuracy of the analyzed algorithms on the SEA_G data set. For the data stream with gradual drifts, the best performance is A-AUE2, followed by AUE2 and HATCL. ADWIN is the worst-performing algorithm.

Fig. 10 shows the prequential accuracy of the algorithms on the SEA_S data set and Fig. 11 displays the performance on the $Hyper_I$. Fig. 12 illustrates on the SEA_{SR} data set of the accuracy of tested algorithms and Fig. 13 reports the performance on the $RanRBF_{GR}$ data set which contains the four gradual recurring drifts. In these four figures, the subsequent drops in the accuracy of ADWIN algorithm. The Average accuracy of the $Shuttle$ data set is shown Fig. 14. NB, ADWIN, ADOB and AWE are the more severely malfunctioning algorithms in this figure.

According to the detailed results from Figs. 3-8 and Figs. 9-14, ADWIN is more poor accuracy than other algorithms. Besides, ADWIN drift detector may not be as good as the performance of ensemble approaches. AUE2 is the overall second-best performance on the data set because of reacting equally well to different types of drift. A-AUE2 approach is the best performing method using HT and HATCL, respectively, as base learner because A-AUE2 reduces the window

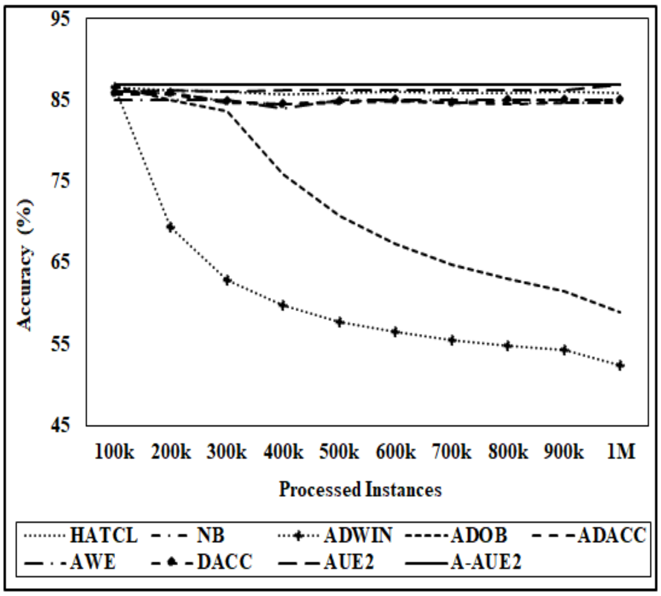


Fig. 9. Classification accuracy of drift detection and handling methods using HATCL as base classifier for SEA_G data set

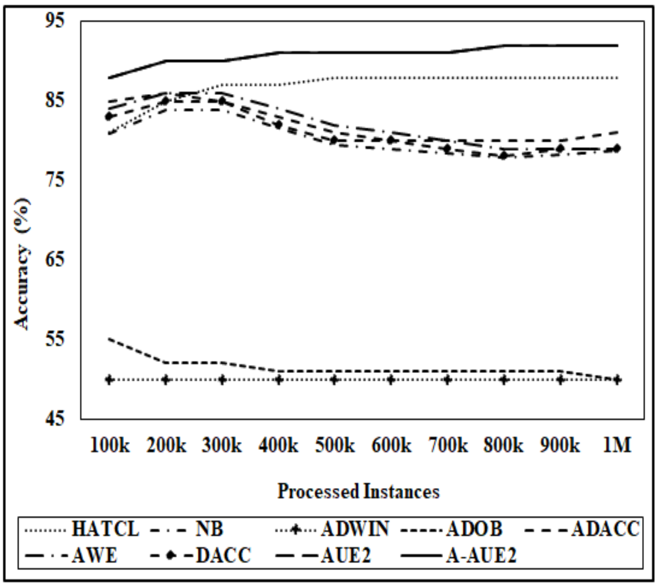


Fig. 11. Classification accuracy of drift detection and handling methods using HATCL as base classifier for $Hyper_I$ data set

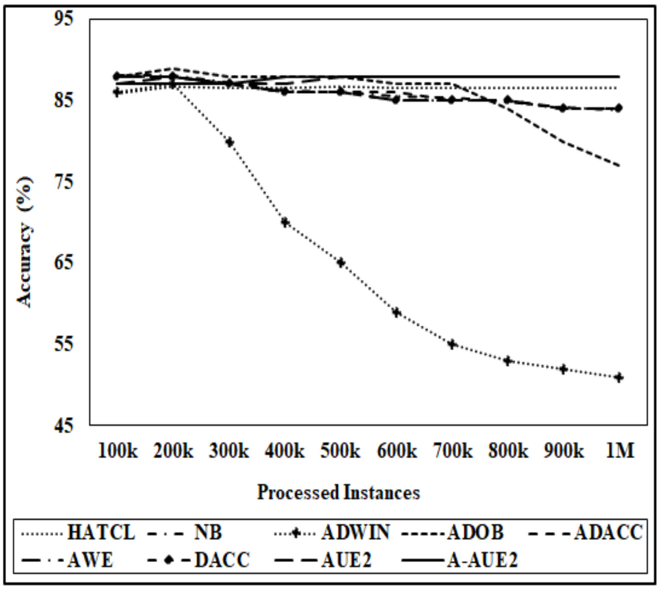


Fig. 10. Classification accuracy of drift detection and handling methods using HATCL as base classifier for SEA_S data set

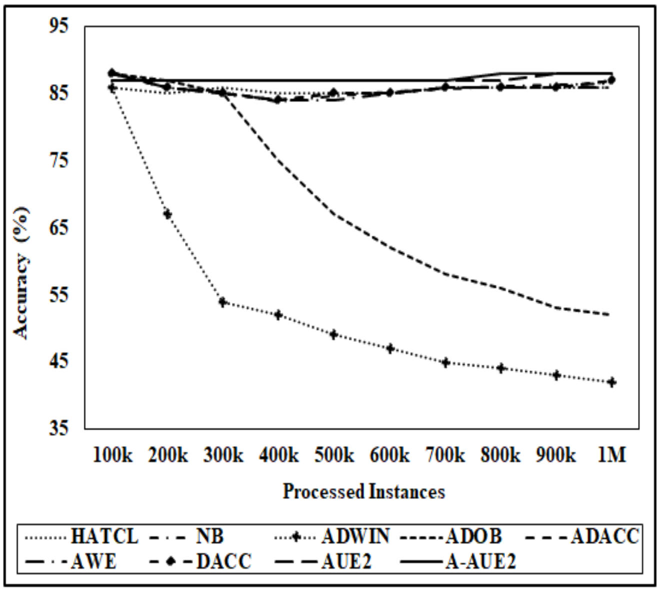


Fig. 12. Classification accuracy of drift detection and handling methods using HATCL as base classifier for SEA_{SR} data set

size problem and handles the noisy samples in data streams.

Table IV presents the results regarding the evaluation of the methods using Precision, Recall, F-Measure (F1) [28], MCC and AUC of drift detection and handling based on HT and HATCL classifiers.

In Table IV, considering the Precision results for HT as base classifier, A-AUE2 obtained better on 5 out of 10 data sets except $RanRBF_{GR}$, $RanTree_{SRF}$, $Usenet2$, $Weather$ and $Airlines$. For Precision results for HATCL as base learner, our proposed A-AUE2 approach obtained the highest results on 5 out of 10 data sets: SEA_G, $Hyper_I$, SEA_{SR}, $RanRBF_{GR}$ and $Shuttle$. Regarding the Precision results in this table, we can conclude that A-AUE2 can improve the Precision results using two base learners for 5 data sets while yields no obvious improvement for 5 data sets.

Similarly, For Recall values of HT as the base classifier, A-AUE2 obtained higher values on 5 out of 10 data sets except $RanRBF_{GR}$, $RanTree_{SRF}$, $Usenet2$, $Weather$ and $Airlines$ in this table. AUE2 outperformed on 2 data sets: $RanRBF_{GR}$ and $Airlines$. For HATCL classifier in this table, our proposed approach achieved better Recall results on 6 out of 10 data sets while ADOB performed better on the other 2 data sets and AUE2 method obtained better on the other one data sets.

In F1 result for HT base classifier, A-AUE2 can perform better on 5 data sets and improve the F1 results of SEA_G, SEA_S, $Hyper_I$, SEA_{SR} and $Shuttle$ data sets in this table. For HATCL base classifier of F1 result, A-AUE2 proposed approach achieved the highest results on 6 out of 10 data sets

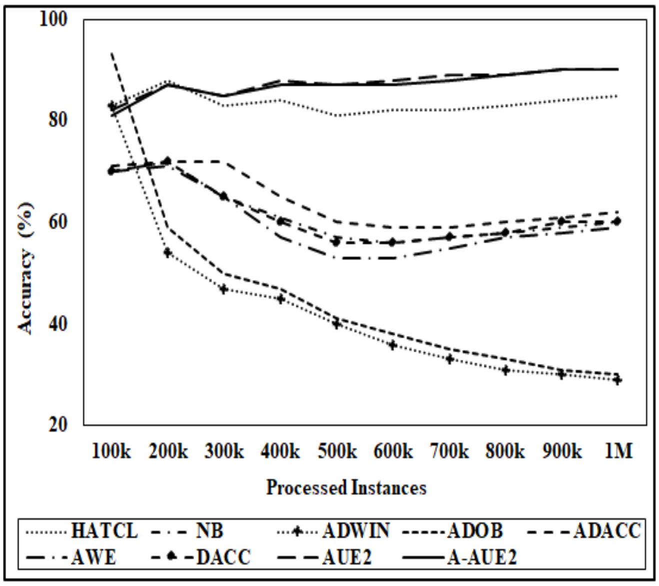


Fig. 13. Classification accuracy of drift detection and handling methods using HATCL as base classifier for $RanRBF_{GR}$ data set

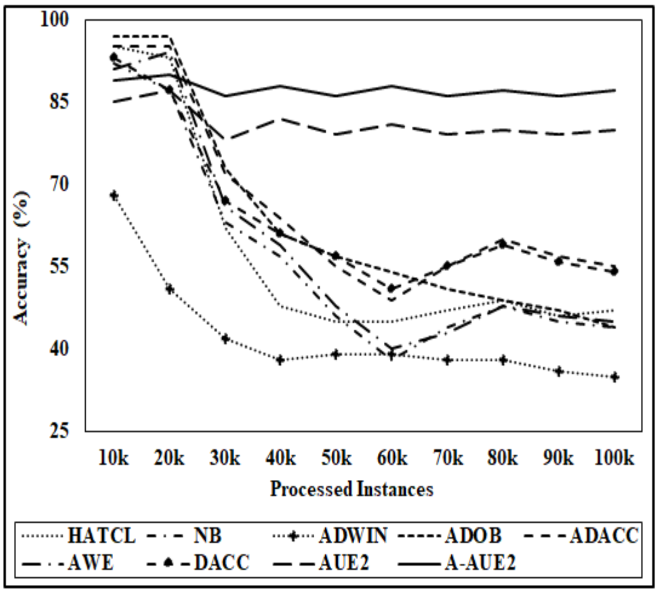


Fig. 14. Classification accuracy drift detection and handling methods using HATCL as base classifier for Shuttle data set

except $RanTree_{SRF}$, $Usenet2$, $Weather$ and $Airlines$.

According to the MCC results for HT as the base classifier, our proposed A-AUE2 approach is more successful on 6 out of 10 data sets than the other drift detection and handling methods. In MCC results for HATCL as base classifier, our proposed approach is outperformed 6 data sets: SEA_G, SEA_S, $Hyper_I$, SEA_{SR}, $RanRBF_{GR}$ and $Shuttle$.

In AUC results of all representative methods by using HT as base learner, our proposed approach obtained the highest results on 5 out of 10 data sets than the other eight comparative methods. For the AUC results of HATCL as base learner, the proposed A-AUE2 approach achieved better results on 6 out of 10 data sets except $RanTree_{SRF}$, $Usenet2$, $Weather$ and $Airlines$.

In order to the above results, we conclude that A-AUE2

TABLE VIII
RESULTS OF FRIEDMAN TEST USING HT AND HATCL BASE CLASSIFIERS

HT		HATCL	
Evaluation Methods	F_F	**Evaluation Methods**	F_F
Average accuracy	15.65	Average accuracy	14.39
Precision	5.15	Precision	4.96
Recall	7.88	Recall	10.17
F1	8.29	F1	10.10
MCC	9.02	MCC	12.60
AUC	8.62	AUC	12.36

can increase the average Precision, Recall, F1, MCC and AUC for most of the data sets. In this paper, our proposed approach (A-AUE2) of Average accuracy, Precision, Recall, F1, MCC and AUC is better than AUE2 using HT and HATCL, respectively, as base learner because our proposed approach uses the noise filtering method (KNN) over each window and the adaptive windowing method (BSS).

B. Statistical Analysis

In Table V, the Average accuracy using HT as base classifier is ranked the ascending order in the computation of Friedman Test over all data sets. A rank of 1 is assigned to the highest accuracy, a rank of 2 is assigned to the next highest accuracy, and so on. When the accuracy values of two or more are the same, the position ranks of these same values are aggregated as the sum value, then this sum value is divided by the count of the same accuracy value. For example, if the fifth and sixth ranks are equal in accuracy values, each is assigned a rank of 5.5. In Table VI and Table VII, the average ranks of Precision, Recall, F1, MCC and AUC are calculated such as Table V.

Fig. 15 and Fig. 16 present the graphical representation of the average ranks of Friedman Test obtained by the prequential classification Average accuracy, Precision, Recall, F1, MCC and AUC using two base learners. For the clear representation of the graphs, the notations used in figures are presented in Table VI and Table VII. As a result of Table VI, the proposed A-AUE2 approach is the best rank in Average accuracy, Recall, F1, MCC and AUC except Precision. In this table, the average rank of Precision for A-AUE2 and AUE2 are equal. According to the results of Table VII, our proposed approach is the best average rank in Average accuracy, Precision, Recall, F1, MCC and AUC.

Table VIII illustrates the value of F_F from the Friedman Test for Average accuracy, Precision, Recall, F1, MCC and AUC. The null hypothesis is rejected for all evaluation methods, respectively because the critical value for $\propto$ =0.05 is 2.07. As a result, there are significant differences in these evaluation methods of the tested algorithms.

The statistical significance tests of our proposed A-AUE2 approach are presented using Wilcoxon's Signed Rank Test and the detailed results are shown in Table IX. In this table, the $Win-Tie-Loss$ statistics by summarizing the results, all positive-sum ranks (W^+), all negative-sum ranks (W^-) and P_{value} of Wilcoxon's Signed Rank Test between the pairs of A-AUE2 and comparative drift detection methods based on Table II and Table III. The performance of the two methods is significantly different if P_{value} is less than 0.05.

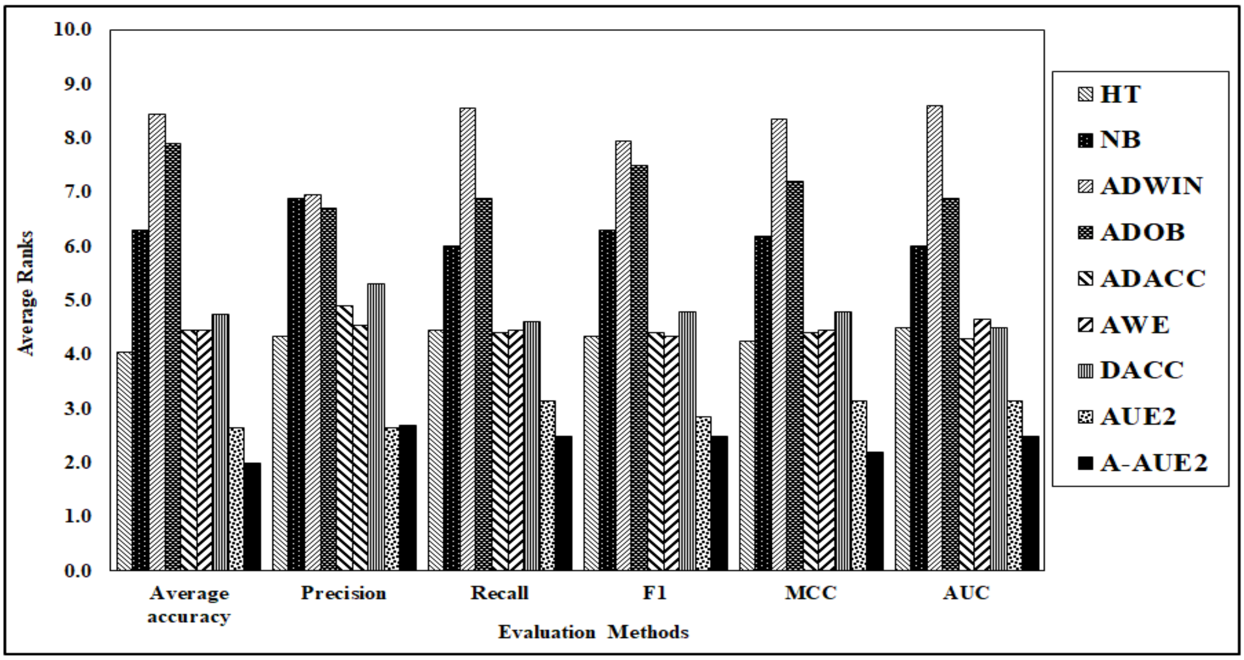


Fig. 15. Average ranks for evaluation methods using drift detection and handling methods based on HT

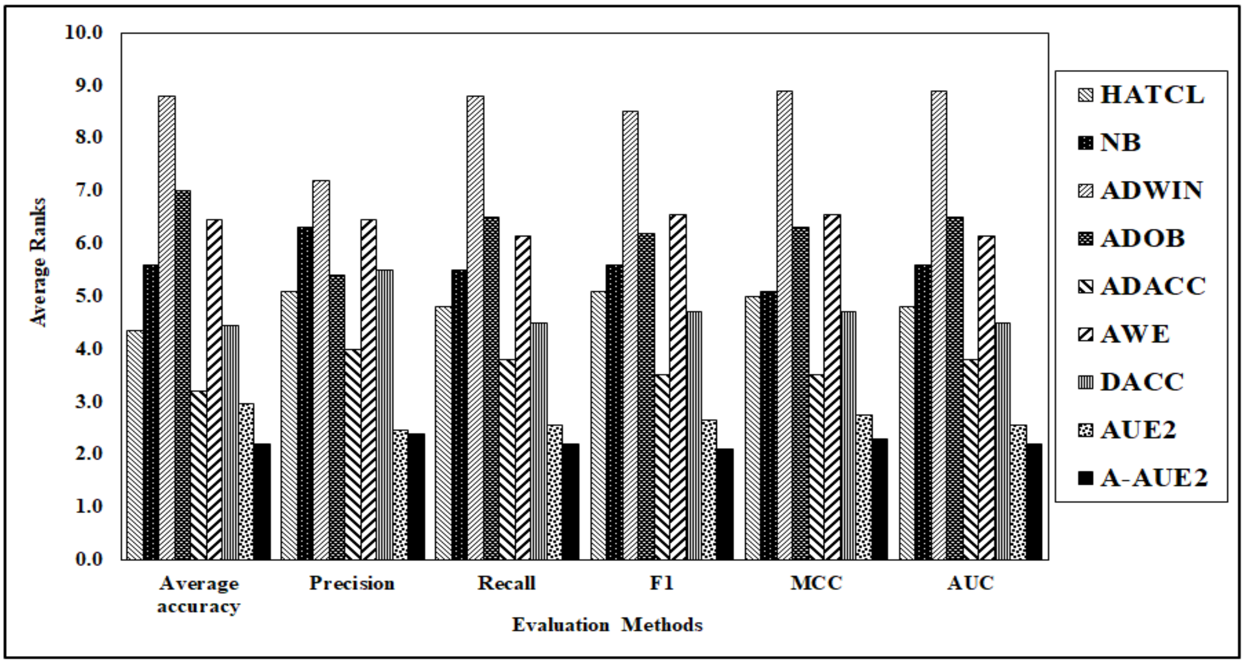


Fig. 16. Average ranks for evaluation methods using drift detection and handling methods based on HATCL

In this Table IX, all positive-sum ranks (W^+) are higher than all negative-sum ranks (W^-) in every pair with both base classifiers. For the P_{value} of Wilcoxon's Signed Rank Test using drift detection and handling based on HT classifier, A-AUE2 method is significantly different in HT, NB, ADWIN, ADOB and AWE but it does not differ with ADACC, DACC and AUE2. For the base classifier of HATCL, our proposed approach performs significantly better than the other compared methods: NB, ADWIN, ADOB, AWE and DACC while HATCL, ADACC and AUE2 are not different with our proposed approach. These results tables show that A-AUE2 is more accurate than all the compared methods with two base learners.

VI. Conclusion and Future Work

An ensemble classifiers algorithm A-AUE2 is proposed in this paper to effectively tackle four types of concept drift. The performance of this research improved by handling the influence of the sizes of the data windows on the performance of the ensemble classifiers. The proposed approach of this paper reduces the window size problem and handles the noisy samples from the data stream using the adaptive windowing method based on Brier Skill Score (BSS) and K-Nearest Neighbors (KNN) based on the noise filtering method.

TABLE IX
WILCOXON'S SIGNED RANK TEST OF AVERAGE ACCURACY FOR DRIFT DETECTION AND HANDLING METHODS USING HT AND HATCL BASE CLASSIFIERS

Hoeffding Tree(HT)					Hoeffding Adaptive Tree Classifier Leaves (HATCL)				
Methods	W^+	W^-	$Win-Tie-Loss$	P_{value}	**Methods**	W^+	W^-	$Win-Tie-Loss$	P_{value}
A-AUE2 vs HT	49	6	9-0-1	**0.028**	A-AUE2 vs HATCL	45	10	8-0-2	0.074
A-AUE2 vs NB	52	3	9-0-1	**0.013**	A-AUE2 vs NB	48	7	9-0-1	**0.037**
A-AUE2 vs ADWIN	54	1	9-0-1	**0.007**	A-AUE2 vs ADWIN	55	0	10-0-0	**0.005**
A-AUE2 vs ADOB	55	0	10-0-0	**0.005**	A-AUE2 vs ADOB	52	3	8-0-2	**0.013**
A-AUE2 vs ADACC	39	16	8-0-2	0.241	A-AUE2 vs ADACC	42	13	7-0-3	0.139
A-AUE2 vs AWE	50	5	9-0-1	**0.022**	A-AUE2 vs AWE	49.5	5.5	9-0-1	**0.025**
A-AUE2 vs DACC	39	16	8-0-2	0.241	A-AUE2 vs DACC	48	7	9-0-1	**0.037**
A-AUE2 vs AUE2	36	19	8-0-2	0.386	A-AUE2 vs AUE2	41	14	8-0-2	0.169

The performance of our proposed approach is validated with the five-class label predictive metrics (such as F1, Precision, Recall, MCC and AUC criteria). Our results show that A-AUE2 outperformed among the eight state-of-the-art concept drift detection and handling methods with both base learners. The null hypothesis of all evaluation methods is rejected by comparing the average ranks with the critical value in Friedman Test. Moreover, the performance of A-AUE2 approach is significantly improved among the representative methods according to the P_{value} of Wilcoxon Signed Rank Test.

In the future, we intend to expand A-AUE2 approach for the use of the nominal data sets and explore the possibility of adapting the proposed algorithm to work in partially labeled data.

REFERENCES

[1] I. Zliobaite, "Learning under concept drift: An overview," *arXiv preprint arXiv:1010.4784*, 2010.

[2] I. Zliobaite, "Adaptive training set formation," *Doctoral dissertation, Vilnius University*, 2010.

[3] A. Bifet, "Adaptive learning and mining for data streams and frequent patterns," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp55-56, 2009.

[4] J. Gama, P. Medas, G. Castillo and P. Rodrigues, "Learning with drift detection," *In Brazilian symposium on artificial intelligence, Springer, Berlin, Heidelberg*, pp286-295, 2004.

[5] M. Baena-Garcıa, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda and R. Morales-Bueno, "Early drift detection method," *In Fourth international workshop on knowledge discovery from data streams*, vol. 6, pp77-86, 2006.

[6] K. Nishida and K. Yamauchi, "Detecting concept drift using statistical testing," *In International conference on discovery science, Springer, Berlin, Heidelberg*, pp264-269, 2007.

[7] I. Khamassi and M. Sayed-Mouchaweh, "Drift detection and monitoring in non-stationary environments," *IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pp1-6, 2014.

[8] D. Jankowski, K. Jackowsk and B. Cyganek, "Learning decision trees from data streams with concept drift," *Procedia Computer Science*, vol. 80, pp1682-1691, 2016.

[9] X. Wang, B. Chen and F. Chang, "A classification algorithm for noisy data streams with concept-drifting," *J. Comput. Inf. Syst*, vol. 7, no. 12, pp4392-4399, 2011.

[10] D. R. de Lima Cabral and R. S. M. de Barros, "Concept drift detection based on Fisher's Exact test," *Information Sciences*, vol. 442, pp220-234, 2018.

[11] S. Ren, B. Liao, W. Zhu and K. Li, "Knowledge-maximized ensemble algorithm for different types of concept drift," *Information Sciences*, vol. 430, pp261-281, 2018.

[12] H. Wang, W. Fan, P. S. Yu and J. Han, "Mining concept-drifting data streams using ensemble classifiers," *In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp226-235, 2003.

[13] B. Krawczyk and A. Cano, "Online ensemble learning with abstaining classifiers for drifting and noisy data streams," *Applied Soft Computing*, vol. 68, pp677-692, 2018.

[14] O. A. Mahdi, E. Pardede and J. Cao, "Combination of information entropy and ensemble classification for detecting concept drift in data stream," *In Proceedings of the Australasian Computer Science Week Multiconference*, pp1-5, 2018.

[15] A. Jadhav, and L. Deshpande, "An efficient approach to detect concept drifts in data streams," *In 2017 IEEE 7th International Advance Computing Conference (IACC)*, pp28-32, 2017.

[16] M. Dehghan, H. Dehghan and P. ZareMoodi, "A novel concept drift detection method in data streams using ensemble classifiers," *Intelligent Data Analysis*, vol. 20, no. 6, pp1329-1350, 2016.

[17] Y. Sun, H. Shao and S. Wang, "Efficient ensemble classification for multi-label data streams with concept drift," *Information*, vol. 10, no. 5, pp158, 2019.

[18] D. Brzezinski and J. Stefanowski, "Reacting to different types of concept drift: The accuracy updated ensemble algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp81-94, 2013.

[19] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *University of California, Santa Cruz, Computer Research Laboratory*, pp256-261, 1989.

[20] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: An ensemble method for drifting concepts," *Journal of Machine Learning Research*, vol. 8, no. Dec, pp2755-2790, 2007.

[21] N. Candogan Yossef, R. Van Beek, A. Weerts, H. Winsemius and M. F. Bierkens, "Skill of a global forecasting system in seasonal ensemble streamflow prediction," *Hydrology and Earth System Sciences*, vol. 21, no. 8, pp4103-4114, 2017.

[22] M. M. Nwe and K. T. Lynn, "Effective Resampling Approach for Skewed Distribution on Imbalanced Data Set," *IAENG International Journal of Computer Science*, vol. 47, no. 2, pp234-249, 2020.

[23] J. Diaz-Rozo, C. Diaz-Rozo and P. Larrañaga, "Clustering of data streams with dynamic gaussian mixture models: an IoT application in industrial processes," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp3533-3547, 2018.

[24] A. Bifet, G. Holmes, R. Kirkby and B. Pfahringer, "Moa: Massive online analysis," *Journal of Machine Learning Research*, vol. 11, no. May, pp1601-1604, 2010.

[25] G. Hulten, L. Spencer and P. Domingos, "Mining time-changing data streams," *In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp97-106, 2001.

[26] P. Langley, W. Iba and K. Thompson, "An analysis of Bayesian classifiers," *In Aaai*, pp223-228, 1992.

[27] A. Bifet, J. Read, I. Žliobait e, B. Pfahringer and G. Holmes, "Pitfalls in benchmarking data stream classification and how to avoid them," *Machine Learning and Knowledge Discovery in Databases, Springer*, pp465–479, 2013.

[28] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp861-874, 2006.

[29] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, pp1-30, 2006.

[30] A. Tsymbal, "The problem of concept drift: definitions and related work," *Computer Science Department, Trinity College Dublin*, vol. 206, no. 2, pp58, 2004.

[31] J. H. McDonald, "Handbook of Biological Statistics (3rd ed.)," *Sparky House Publishing, Baltimore, Maryland*, 2014.

[32] G. Jaber, A. Cornuéjols and P. Tarroux, "A new on-line learning method for coping with recurring concepts: the ADACC system,"

International conference on neural information processing, Springer, Berlin, Heidelberg, pp595-604, 2013.
[33] N. C. Oza and S. Russell, "Online ensemble learning," *University of California, Berkeley*, 2001.
[34] F. Wilcoxon, "Some rapid approximate statistical procedures," *Annals of the New York Academy of Sciences*, vol. 52, no. 1, pp808-814, 1950.
[35] A. A. Bradley, S. S. Schwartz and T. Hashino, "Sampling uncertainty and confidence intervals for the Brier score and Brier skill score," *Weather and Forecasting*, vol. 23, no. 5, pp992-1006, 2008.
[36] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Information Fusion*, vol. 37, pp132-156.
[37] B. R. Prasad and S. Agarwal, "Stream data mining: platforms, algorithms, performance evaluators and research trends," *International journal of database theory and application*, vol. 9, no. 9, pp201-218, 2016.
[38] M. M. Gaber, A. Zaslavsky and S. Krishnaswamy, "A survey of classification methods in data streams," *In Data streams, Springer, Boston, MA*, pp39-59, 2007.
[39] Y. Sun, Z. Wang, H. Liu, C. Du and J. Yuan, "Online ensemble using adaptive windowing for data streams with concept drift," *International Journal of Distributed Sensor Networks*, vol. 12, no. 5, pp4218973, 2016.
[40] J. P. Barddal, H. M. Gomes, F. Enembreck, B. Pfahringer and A. Bifet, "On dynamic feature weighting for feature drifting data streams," *In Joint european conference on machine learning and knowledge discovery in databases, Springer, Cham*, pp129-144, 2016.
[41] S. G. T. de Carvalho Santos, P. M. G. Júnior, G. D. dos Santos Silva and R. S. .M. de Barros, "Speeding up recovery from concept drifts," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, Berlin, Heidelberg*, pp179-194, 2014.
[42] G. Jaber, A. Cornuéjols and P. Tarroux, "Online learning: Searching for the best forgetting strategy under concept drift," *In International Conference on Neural Information Processing, Springer, Berlin, Heidelberg*, pp400-408, 2013.
[43] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Structure*, vol. 405, no. 2, pp442-451, 1975.
[44] R. S. M. de Barros, J. I. G. Hidalgo and D. R. de Lima Cabral, "Wilcoxon rank sum test drift detector," *Neurocomputing*, vol. 275, pp1954-1963, 2018 .
[45] J. Liu, Q. Miao, Y. Sun, J. Song and Y. Quan, "Fast structural ensemble for one-class classification," *Pattern Recognition Letters*, vol. 80, pp179-187, 2016.
[46] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly weather review*, vol. 78, no. 1, pp1-3, 1950.
[47] S. D. Bay, "Nearest neighbor classification from multiple feature subsets," *Intelligent data analysis*, vol. 3, no. 3, pp191-209, 1999.
[48] M. Bilal, H. Israr, M. Shahid and A. Khan, "Sentiment classification of Roman-Urdu opinions using Naïve Bayesian, Decision Tree and KNN classification techniques," *Journal of King Saud University-Computer and Information Sciences*, vol. 28, no. 3, pp330-344, 2016.
[49] K. V. Kokilam and D. P. P. Latha, "Ensemble Method to classify multi class with concept drift," *In Journal of Physics: Conference Series, IOP Publishing*, vol. 1706, no. 1, pp012151, 2020.
[50] 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, pp150-166, 2018.