A Conformal Martingales Ensemble Approach for addressing Concept Drift

Charalambos Eliades ST009072@STUD.FREDERICK.AC.CY
Computational Intelligence (COIN) Research Lab
Frederick University, Cyprus
Harris Papadopoulos H.PAPADOPOULOS@FREDERICK.AC.CY
Computational Intelligence (COIN) Research Lab
Frederick University, Cyprus

Editor: Harris Papadopoulos, Khuong An Nguyen, Henrik Boström and Lars Carlsson

Abstract

We propose an ensemble learning approach to tackle the problem of concept drift (CD) in data-stream classification. Accurately detecting the change point in the distribution is insufficient to ensure precise predictions, particularly when the selection of a representative training set is challenging or computationally expensive. More specifically, we employ an ensemble of ten classifiers that use a majority voting mechanism to make predictions. To promote diversity among models, we train each on a different number of instances, resulting in different sequences of p-values and construct an Inductive Conformal Martingale (ICM) for each one. When the ICM algorithm detects a change point in the corresponding p-value sequence, we perform a retraining process of the corresponding classifier. We evaluate the performance of our proposed methodology on four benchmark datasets and compare it to existing methods in the literature. Our experimental results show that the proposed approach exhibits comparable and in some cases better accuracy than two state-of-the-art algorithms.

Keywords: Conformal, Martingales, Exchangeability,Drift

1. Introduction

This study is an extension of our previous work (Eliades and Papadopoulos, 2022), where we used ICM with a newly proposed betting function to quickly detect the point at which a CD occurs and retrain the model to regain accuracy. The speed of change point detection however, is not the only aspect affecting the resulting accuracy. The selection of appropriate instances for the training and retraining of the model is equally if not more important. Selecting a representative training set is a challenging problem that might require a significant computational effort to be performed. To deal with this issue, we choose to train ten different models with a varying number of training instances to promote diversity among models. Each model produces a different p-value sequence and in each sequence, an ICM is applied for change detection. When a change is detected,we construct a new training set and retrain the corresponding classifier. The points of change detected by different models do not necessarily coincide. The fact that the change points do not coincide means that each model is trained on a different subset of the training set, making us able to have predictions for a larger number of instances; even in the case of a false alarm, the remaining models will continue to make predictions thus minimizing the number of instances that a prediction label is not available. Additionally if some model misses or detects a change late, the rest of the models will mitigate the effect this will have on accuracy.

CD corresponds to a change in the underlying data generating mechanism, resulting in a loss of classification performance. Using ICM on the calculated p-values of each classifier in the ensemble, we detect violations of the exchangeability assumption (EA) for a pre-specified significance level and retrain the corresponding classifier of the ensemble to regain performance.

Formally a CD can be defined as follows: Given a data stream $S = \{(x_0, y_0), (x_1, y_1), \dots, \}$, where x_i is a feature vector and y_i a label; if S can be divided into two sets $S_{0,t} = \{(x_0, y_0), \dots, (x_t, y_t)\}$ and $S_{t+1,\dots} = \{(x_{t+1}, y_{t+1}), \dots\}$, such that $S_{0,t}$ and $S_{t+1,\dots}$ are generated by two different distributions, then a CD occurred at timestamp $t + 1$. This can be extended to any number of CDs with any number of different distributions.

CD can be produced from three sources. Recall that $f_{X,Y,t} = f_{X|Y,t} \cdot f_{Y,t}$ and $f_{X,Y,t+1} = f_{X|Y,t+1} \cdot f_{Y,t+1}$, where $f_{X,Y,t}$ is the joint probability density function of a pair (x, y) at time t . To have a change in the joint distribution of (X, Y), one of the following might have happened: (a) $f_{X,Y,t} = f_{X|Y,t}$ and $f_{Y,t} \neq f_{Y,t+1}$, in this case we have a change in the label's distribution while the decision boundaries remain unchanged, this has also been considered as label shift (Vovk, 2020) and virtual drift (Bagui and Jin, 2020) (b) $f_{X|Y,t} \neq f_{X|Y,t+1}$ and $f_{Y,t} = f_{Y,t+1}$ here the decision boundaries change and lead to decrease in accuracy, it has also been considered as actual drift (Bagui and Jin, 2020) or concept shift (Vovk, 2020) and (c) $f_{X|Y,t} \neq f_{X|Y,t+1}$ and $f_{Y,t} \neq f_{Y,t+1}$ which is a combination of the two previously mentioned sources.

CD types can be classified into four categories: (a) sudden drift, where the data generating mechanism changes instantly (b) gradual drift, where the data distribution is replaced with a new one over time (i.e. over time we experience fewer examples belonging to the initial distribution and more belonging to the new distribution), (c) incremental where a new data generating mechanism incrementally replaces the existing mechanism (i.e. each example is generated by a mixture of distributions but over time the impact of the initial distribution disappears) d) reoccurring drift when a previously seen data distribution reappears (Lu et al., 2019),(Bagui and Jin, 2020) .

The CD detection algorithms can be classified into three categories based on the statistics they apply (Lu et al., 2019). The first category is the 'Error rate-based algorithms', which monitor increases or decreases in the online error rate, if these changes are statistically significant, a drift alarm is raised. The second and biggest category is the 'Data Distribution-based'; here the algorithms quantify the dissimilarity between the historical data and the new data. A distance function is used to measure the dissimilarity between the two distributions and a statistical hypothesis test with respect to a significance level determines if a CD occurs. The last category 'Multiple Hypothesis Test', applies similar techniques to the ones mentioned above but employs multiple hypothesis tests to determine the presence of a CD. They can be divided into two groups: parallel hypothesis tests and hierarchical multiple hypothesis tests; for more information refer to (Lu et al., 2019). In this study our CD detection algorithm belongs to the second category.

The rest of the paper starts with an overview of related work on CD in Section 2. Section 3 gives a brief overview of the ideas behind ICM. Section 4 describes the proposed approach. Section 5 presents the experimental setting and performance measures used in our evaluation and reports the obtained experimental results. Finally, Section 6 gives our conclusions and plans for future work.

2. Related Work

Given the vast amount of research on this topic we will present only a few works related to the method we follow.

2.1. Conformal Martingales

In this section, we present the work done by several researchers on testing the exchangeability assumption using Conformal Martingales.

Vovk et al. (2003) proposed a way of testing exchangeability online based on Conformal Prediction (CP) and CM. This technique consists of calculating a sequence of p-values by applying conformal prediction. The p-values are calculated online, where the p-value of each new example is calculated from the new and previously seen examples. After the p-values are calculated, a BF is applied to each p-value and the product of the BF outputs is the value of the Martingale. When the value M of the Martingale becomes large enough, we can reject the exchangeability assumption at a significance level $1/M$. This method can be used to test if a dataset satisfies the EA and for change point detection in time series and consequently for CD. Some related techniques are presented below.

Ho (2005) implemented a CM based on a simple betting mixture function extending the idea of detecting exchangeability online to detect concept changes in time-varying data streams. Two martingale tests were implemented based on: (i) martingale values and (ii) the martingale difference. The Martingale was calculated using the mixture BF.

Fedorova et al. (2012) tested the exchangeability of data on two datasets, USPS and Statlog Satellite data. The data is tested online, i.e. the examples arrive one by one and then the value of the CM is calculated, which is a valid measure indicating if the EA should be rejected. In this article the authors used a density estimator of the observed p-values as a BF. The kernel density estimation has been employed and it has been shown to outperform the simple mixture BF.

Volkhonskiy et al. (2017) implemented an Inductive version of CM to detect when a change occurs in a time series. In this study, the underlying model is trained on the first observations of the time sequence. All the nonconformity scores are calculated via the underlying model. The authors used several BF and showed that the pre-computed kernel BF provides the most efficient results (i.e. lowest mean delay). They tested their methods on synthetic datasets and showed that their results are comparable with those of many other methods such as CUSUM, Shiryaev-Roberts and Posterior Probability statistics.

Ho et al. (2019) proposed a real time novel martingale-based approach driven by Gaussian process regression (GPR) to predict and detect anomalous flight behaviour as observations arrive one by one. The authors use multiple CM tests to reduce the number of false alarms and the delay time required for anomaly detection. Here again, the Martingale was calculated using the mixture BF.

2.2. Concept drift

In this subsection we present some related work to the CD problem. We start by giving two surveys: Lu et al. (2019) give an overview of over 130 high quality publications in CD related research areas, they list and discuss ten popular synthetic datasets and 14 publicly available benchmark datasets used for evaluating the performance of learning algorithms dealing with CD.

Bagui and Jin (2020) surveyed works dealing with CD, and they presented a comprehensive study of public synthetic and real datasets that can be used to benchmark such a problem. They review the different CD types and approaches used to handle such changes in the data.

As mentioned earlier CD detection algorithms can be classified in three categories based on the statistics they apply (Lu et al., 2019). The first category is the 'Error rate-based algorithms', , the second and biggest category is the 'Data Distribution-based', the last category 'Multiple Hypothesis Test'. In the remainder of this subsection we provide some work related to these categories.

2.2.1. Error rate based methods

Wang et al. (2003) proposed a new method called Accuracy Weighted Ensemble (AWE). This ensemble based method assigns weights to the base classifiers based on the classification error.

Liao and Dai (2014) suggested a hybrid method called Accuracy and Growth rate updated Ensemble (AGE), which combines the strengths of single classifier and ensemble methods. AGE is designed using the geometric mean of weights and growth rates of models, which enhances its adaptability to different types of concept drifts. The results of the experiments indicate that AGE generally outperforms its competitors in terms of accuracy.

Kolter and Maloof (2007) presented an ensemble method for CD that creates, weights or removes online learners based on their performance. The authors call their method Dynamic Weighted Majority (DWM) and combine it with Naive Bayes (DWM-NB).

Elwell and Polikar (2011) proposed an ensemble-based approach called Learn++.NSE. Learn++.NSE is designed for incremental learning of concept drift in nonstationary environments. The proposed algorithm learns from consecutive batches of data without any assumptions on the nature or rate of drift and generates a new classifier for each batch of data received. These classifiers are combined using a dynamically weighted majority voting technique based on time-adjusted errors. This mechanism allows the algorithm to effectively integrate current and past classifiers and adapt to changes in the data distribution over time. Learn++.NSE provides a versatile framework for learning in nonstationary environments, which can handle various types of concept drift, such as a constant or variable rate of drift, addition or deletion of concept classes, and cyclical drift.

Fok et al. (2017) used a particle filter-based learning method (PF-LR), for training logistic regression models from evolving data streams. Here the step particles are sampled from the ones that maximize the classification accuracy on the current data batch. Their experiments show that this method performs well, even with relatively small batch sizes. They tested the proposed methods on synthetic and real data sets and found that PF-LR outperforms other state-of-the-art stream mining algorithms on most of the tested data sets.

2.2.2. Data distribution Based

Li et al. (2015) implemented a classification algorithm based on Ensemble Decision Trees for Concept-drifting data streams (EDTC). Extensive studies on synthetic and real streaming databases demonstrate that the EDTC performs very well compared to several known online algorithms based on single and ensemble models.

Liu et al. (2017) focused on concept drift in data stream mining, a measurement called Local Drift Degree (LDD) is proposed to quantify regional density discrepancies between two different sample sets, allowing for the identification of density increased, decreased, and stable regions. The paper also introduces two algorithms, LDD-DIS and LDD-DSDA, which both utilize LDD to address concept drift. LDD-DIS continuously monitors regional density changes using LDD to identify drifted instances within two batches of data. At the same time, LDD-DSDA identifies drifted instances using LDD and adapts the model to handle the drift. Experimental evaluations on three benchmark datasets demonstrate that both algorithms improve accuracy compared to other methods.

Bu et al. (2017) suggested an incremental least squares density difference (LSDD) change detection method, examining the difference between two distributions using two nonoverlapping windows. They tested their method on six synthetic datasets and one real world dataset.

Eliades and Papadopoulos (2021) examined the use of Inductive Conformal Martingales (ICM) combined with the histogram betting function for detecting violation of the EA and therefore CD.

2.2.3. Multiple Hypothesis Tests

Alippi and Roveri (2008) implemented the just-in-time approach (JIT) to handle recurring concepts effectively. JIT identifies the concept to which examples belong and maintain models representing different concepts in a pool. It shows that exploiting information acquired in the past helps.

Alippi et al. (2017) proposed HCDTs (Hierarchical Change-Detection Tests), an approach to address concept drift using a two-layer hierarchical architecture. The detection layer, which is responsible for promptly detecting possible changes in the data generating mechanism, can be any low-complexity drift detection method. Upon detecting a change, the validation layer is activated to perform a more advanced analysis on recently acquired data, which helps reduce false alarms. HCDTs can automatically reconfigure after having detected and validated a change, allowing for further detection of departures from the new state of the data-generating process. To design the validation layer, the authors suggest two strategies: estimating the distribution of the test statistics by maximizing the likelihood, and adapting an existing hypothesis test, such as the Kolmogorov-Smirnov test or the Cramer-Von Mises test, to suit the HCDTs architecture.

3. Inductive Conformal Martingales

In this section we describe the basic concepts of ICM and how our nonconformity scores and p-values are calculated.

3.1. Data Exchangeability

Let $(Z_1, Z_2, \dots)$ be an infinite sequence of random variables. Then the joint probability distribution $\mathbb{P}(Z_1, Z_2, \dots, Z_N)$ (where N is a natural number) is exchangeable if it is invariant under any permutation of these random variables. The joint distribution of the infinite sequence $(Z_1, Z_2, \dots)$ is exchangeable if the marginal distribution of $(Z_1, Z_2, \dots, Z_N)$ is exchangeable for every N. Testing if the data is exchangeable is equivalent to testing if is independent and identically distributed (i.i.d.); this is an outcome of de Finetti's theorem (Schervish, 1995): any exchangeable distribution on the data is a mixture of distributions under which the data is i.i.d.

3.2. Exchangeability Martingale

A test exchangeability Martingale is a sequence of random variables $(S_1, S_2, S_3, \dots)$ being equal to or greater than zero that keep the conditional expectation $\mathbb{E}(S_{n+1}|S_1, \dots, S_n) = S_n$.

To understand how a martingale works, consider a fair game where a gambler with infinite wealth follows a strategy based on the distribution of the events in the game. The gain acquired by the gambler can be described by the value of a Martingale. I.e. Ville's inequality (Ville, 1939) indicates that the probability of having high profit (C) would be small, $\mathbb{P}\{\exists n : S_n \geq C\} \leq 1/C$.

According to Ville's inequality (Ville, 1939) for the case of the EA, a large final value of the Martingale suggests rejection of the assumption with a significance level equal to the inverse of the Martingale value. I.e. a Martingale value such as 10 or 100 rejects the hypothesis of exchangeability at 10% or 1% significance level, respectively.

3.3. Calculating Non-conformity Scores and P-values

Let $\{z_1, z_2, \dots\}$ be a sequence of examples, where $z_i = (x_i, y_i)$ with x_i an object given in the form of an input vector, and y_i the label of the corresponding input vector. The CM approach generates a sequence of p-values corresponding to the given sequence of examples and then calculates the martingale as a function of these p-values. As mentioned in Section 1, this work employs CM's computationally efficient inductive version. ICM uses the first k examples $\{z_1, z_2, \dots, z_k\}$ in the sequence to train a classification algorithm, which it then uses to generate the p-values for the next examples. Consequently, it starts checking for violations of the EA from example z_{k+1} on, i.e. the sequence $\{z_{k+1}, z_{k+2}, \dots\}$.

Our aim is to examine how strange or unusual a new example $z_j \in \{z_{k+1}, z_{k+2}, \dots\}$ is compared to the training examples. To make this possible, we define a function $A(z_i, \{z_1, \dots, z_k\})$, where $i \in \{k+1\ \dots\}$, called a nonconformity measure (NCM) that assigns a numerical value α_i to each example z_i, called nonconformity score (NCS). The NCM is based on the trained underlying classification algorithm. The bigger the NCS value of an example, the less nonconforming it is with $\{z_1, \dots, z_k\}$ with respect to the underlying algorithm.

For every new example z_j we generate the sequence $H_j = \{\alpha_{k+1}, \alpha_{k+2}, \dots, \alpha_{j-1}, \alpha_j\}$ to calculate its p-value. Note that the NCSs in H_j are calculated when the underlying algorithm is trained on $\{z_1, z_2, \dots, z_k\}$. Given the sequence H_j we can calculate the corresponding p-value (p_j) of the new example z_j with the function:

$$p_j = \frac{|\{\alpha_i \in H_j | \alpha_i > \alpha_j\}| + U_j \cdot |\{\alpha_i \in H_j | \alpha_i = \alpha_j\}|}{j-k}, \tag{1}$$

where α_j is the NCS of the new example and α_i is the NCS of the i^{th} element in the example sequence set and U_j is a random number from the uniform distribution (0,1). For more information, refer to Vovk et al. (2003).

3.4. Inductive Conformal Martingales

An ICM is an exchangeability test Martingale (see Subsection 3.2), which is calculated as a function of p-values such as the ones described in Subsection 3.3.

Given a sequence of p-values $(p_1, p_2, \dots)$ the martingale S_n is calculated as:

$$S_n = \prod_{i=1}^{n} f_i(p_i) \tag{2}$$

where $f_i(p_i) = f_i(p_i|p_1, p_2, \dots, p_{i-1})$ is the betting function (Vovk et al., 2003).

The betting function should satisfy the constraint: $\int_0^1 f_i(p)dp = 1, f_i(p) \geq 0$ and also the S_n must keep the conditional expectation: $\mathbb{E}(S_{n+1}|S_0, S_1, \dots, S_n) = S_n$.

The integral $\int_0^1 f_i(p)dp$ equals to 1 because $f_i(p)$ is the p-values $(p_1, p_2, \dots, p_{i-1})$ density estimator. We also need to prove that $\mathbb{E}(S_{n+1}|S_0, S_1, \dots, S_n) = S_n$ under any exchangeable distribution.

Proof $\mathbb{E}(S_{n+1}|S_0, S_1, \dots, S_n) = \int_0^1 \prod_{i=1}^{n} f_i(p_i) \cdot f_{n+1}(p)dp = \prod_{i=1}^{n} f_i(p_i) \cdot \int_0^1 f_{n+1}(p)dp = \prod_{i=1}^{n} f_i(p_i) = S_n$ ■

Using equation (2), it is easy to show that $S_n = S_{n-1} \cdot f_n(p_n)$, which allows us to update the martingale online. Let us say that the value of S_n equals M, then Ville's inequality (Ville, 1939) suggests that we can reject the EA with a significance level equal to $1/M$.

Note that we can calculate equation (2) in the logarithmic scale to deal with precision issues.

4. Proposed Approach

This section describes the proposed approach, which uses an ensemble that consists of 10 classifiers. Each model is trained on a varying number of instances to promote diversity. For each model, a different series of p-values are calculated, and to detect a change point in the data generating mechanism, an ICM is employed, which uses the Cautious Betting Function on top of the histogram and the kernel density estimators (KDE) as in Eliades and Papadopoulos (2022). When a change point is detected at a specific point, we retrain this single model in the ensemble.

4.1. Cautious Betting Function

Here we describe the Cautious Betting Function. An issue of the CM and ICM is that they might need much time to recover from a value very close to zero (Volkhonskiy et al., 2017). This betting function avoids betting (i.e. $h_n = 1$) when insufficient evidence is available to reject the EA, thus keeping the value of S_n from getting close to zero and reducing the time needed to detect a CD. The following theorem supports the decision to choose this betting function.

Theorem 1 *When the distribution of the p-values is uniform. for any betting function other than $f = 1$ then $S_\infty \equiv 0$.*

For a proof refer to (Eliades and Papadopoulos, 2022). The proposed betting function h_n is defined as follows:

$$h_n(x) = \begin{cases} 1 & \text{if } S1_{n-1}/\min_k S1_{n-k} \leq \epsilon \\ f_n & \text{if } S1_{n-1}/\min_k S1_{n-k} > \epsilon \end{cases} \tag{3}$$

with $S1_n = \prod_{i=1}^{n} f_i(p_i)$, $k \in 1, \ldots, W$, $\epsilon > 0$ and $W \in \{1, \ldots, n-1\}$. Here ϵ and W are parameters of the proposed betting function and f_n is a betting function or density estimator.

As mentioned in Section 3.2, the operation of a CM could resemble that of a gambler who makes decisions based on the event distributions in a game. To illustrate the functioning of this betting system, let us consider two players. The first player's strategy is based on the density estimator f_n, a histogram or kernel estimator in this study. The second player observes the first player's performance in the game during a time window $[n-k, n-1]$ and calculates the maximum profit the first player could have made by starting to play when $S1_i$ was at its minimum, i.e., $S1_{n-1}/\min_k S1_{n-k}$. If the first player incurs losses or has very small profits, then the second player will not bet, i.e., h_n will be set to 1. However, if the first player could make a profit greater than a certain threshold, the second player will bet by setting $h_n = f_n$.

4.2. Density estimators

Here we present and describe the density estimators we have used to combine with the Cautious betting function.

4.2.1. Kernel Estimator

The KDE is calculated as:

$$\hat{f_n}(x) = \frac{1}{nh} \sum_{i=n-L+1}^{n} k(\frac{|x - x_i|}{h}) \tag{4}$$

where h is a bandwidth parameter and k is the simple Gaussian:

$$k(z) = \frac{1}{\sqrt{2\pi}} \exp(-\tfrac{1}{2}z^2). \tag{5}$$

While calculating the KDE, we have used Silverman's "rule of thumb" (Silverman, 1986) for bandwidth selection. We have used the reflection method as in (Fedorova et al., 2012) to improve performance for points near the bounds [0,1]. Also, to eliminate the risk of having an x_0 with $\hat{f}_n(x_0) = 0$ leading to martingale values equal to zero, we add a negligible constant to $\hat{f}_n(x)$. Because this constant is set to be extremely small (10^{-10}) it does not disturb the performance of KDE. The integral of $A = \int_0^1 (f_n(x) \hat{+} 10^{-10})dx \approx 1$ thus practically, there is no need to multiply the integral with any constant to force equality to 1.

4.2.2. Histogram Estimator

Compared to KDE this estimator is faster (Eliades and Papadopoulos, 2021) and needs less computational effort to tune. The p-values $p_i \in [0, 1]$, so we partition $[0, 1]$ into a predefined number of bins k and calculate the frequency of the observations that lie in each bin. Dividing these frequencies by the total number of observations and multiplying it by the number of bins gives us the histogram estimator.

Let us take a fixed number of bins κ this will partition $[0, 1]$ into $B_1 = [0, 1/\kappa)$, $B_2 = [1/\kappa, 2/\kappa)$,..., $B_{\kappa-1} = [(\kappa-2)/\kappa, (\kappa-1)/\kappa)$ and $B_\kappa = [(\kappa-1)/\kappa, 1]$. Then for a p-value $p_n \in B_j$ the density estimator will be equal to:

$$\hat{f}_n(p_n) = \frac{n_j.\kappa}{n-1}, \tag{6}$$

Where $n-1$ is the number of p-values seen so far and n_j is the number of p-values belonging to B_j. Note that when n is small it is possible that $\exists x : \hat{f}_n(x) = 0$, in that case until a sufficient number of observations arrives we reduce the number of bins κ by 1, the reduction of κ is repeated until $\nexists x : \hat{f}_n(x) = 0$.

4.3. Detecting CD using ICM

In order to detect a CD at a pre-specified significance level δ, the Martingale value must exceed $1/\delta$, which leads to the rejection of the EA. This process is summarized in Algorithm 1. Specifically, if the Martingale value S_k at a given point k exceeds 100, a CD is detected with a significance level of 1%, where L denotes the number of p-values that our estimator uses.

4.4. Ensemble ICM

In our previous study (Eliades and Papadopoulos, 2022), we used a single classifier and ICM to detect CDs. When a CD was detected, we waited until a pre-specified number of instances arrived to be used as training set. In our current methodology for addressing the CD problem, we train ten classifiers using a predetermined number of instances $k = \{100, \ldots, 1000\}$, due to computational constraints as it is very difficult to calculate an optimum value of k and the ensemble size, since it depends on the characteristics of each dataset. Each classifier is trained with a different number of instances, and as new observations arrive, a distinct sequence of p-values is calculated for each classifier. We apply a different ICM to each sequence of p-values to detect possible CDs. If a CD is detected, the affected classifier stops making predictions.

```
Data: Training set $\{z_1, z_2, \ldots, z_k\}$, Test set$\{z_{k+1}, \ldots, z_n\}$, significance level $\delta$
Initialize $S_1 = 1$
for $i$=1,...,$n$-$k$ do
    $\alpha_i = A(z_{k+i}, \{z1, \ldots, z_k\})$
    $p_i = \frac{\#\{j:\alpha_j > \alpha_i\} + U_j \#\{j:\alpha_j = \alpha_i\}}{i}$
    Calculate betting function $h_i = h(p_{i-L}, \ldots, p_{i-1})$
    $S_i = S_{i-1} \cdot B_i(p_i)$
    if $S_i > \frac{1}{\delta}$ then
        Raise an Alarm
    end
end
```

Algorithm 1: Detect CD using ICM

Since a CD begins before it is detected, we go backward in time to construct the new training set. In particular, the new training set consists of the k instances starting from the point that the Martingale has exceeded a threshold r (smaller than the detection threshold); note that, if necessary, we wait until a sufficient number of instances are observed. Specifically, assuming that a CD occurs at an instance with timestamp i, we find the closest timestamp d to i in which the Martingale passes a value equal to $r = \{2, 10, 100\}$. We include instances with indices $\{d, \ldots, d+k\}$ as the training set, where $d = max\{j : S_j^{classifier} < r\} + 1$ and the classifier is retrained at point $d + k$. Adding instances in the training set that arrived before an alarm was raised allows for predictions for as many instances as possible. This entire process is carried out in parallel for each classifier, and a majority vote is used to predict an instance. The entire algorithm is summarized in Algorithm 2. Even in cases where the algorithm raises a false alarm and forces model retraining, the other classifiers can still provide predictions for the corresponding time interval. If the algorithm fails to detect a CD in the p-values produced by a classifier, this classifier might provide misleading predictions. However, since other classifiers have been retrained, they can counterbalance the predictions of the classifier that was not retrained, thereby ensuring that performance is maintained.

It should be noted that calculating the optimum value of k and the number of classifiers is computationally intensive. We opted for a practical approach to overcome these constraints by selecting a range of delay values. Specifically, we considered $k = \{100, \ldots, 1000\}$. Although this range may not guarantee maximum accuracy, it strikes a balance between computational efficiency and assessing the impact of delays on performance for the different datasets used in this study.

5. Experiments and Results

In this section, we conduct an experimental evaluation of the proposed approach (Algorithm 2) and compare its performance to that of existing methods. Specifically, we examine the performance improvement of the ICM ensemble using two synthetic datasets (STAGGER, SEA) and two real datasets (ELEC, AIRLINES). We compare the proposed approach to two state-of-the-art methods from the literature: the DWM method (Kolter and Maloof, 2007) and the AWE method (Wang et al., 2003).

```
Data: Instances $\{z_1, z_2, \dots, z_n\}$, significance level $\delta$, $L$, $k = \{100, 200, \dots, 1000\}$, $r$
Initialize $Modelisupdated[1:10] == False$, $d[1:10] = 1$

for $i$=1,...,$n$ do
    for Classifier=1,...,10 do
        if $Modelisupdated(Classifier) == true$ then
            $\alpha_i^{classifier} = A^{classifier}(z_i, X)$
            $U_i^{classifier} = rand()$ a random number $U_i^{classifier} \in U(0,1)$
            $p_i^{Classifier} = \frac{\#\{j:\alpha_j^{Classifier} > \alpha_i^{Classifier}\} + U_i^{Classifier} \#\{j:\alpha_j^{Classifier} = \alpha_i^{Classifier}\}}{i}$
            Calculate betting function $h_i = h(p_{i-L}^{Classifier}, \dots, p_{i-1}^{Classifier})$
            $S_i^{Classifier} = S_{i-i}^{Classifier} \cdot B_i(p_i^{Classifier})$
            $pred_i^{Classifier}$=prediction for instance $z_i$ from model(classifier)
            if $S_i^{Classifier} > \frac{1}{\delta}$ then
                $Modelisupdated(Classifier) == false$

                $d(Classifier) = max\{i : S^{Classifier} < r\} + 1$

            end
        end
        else if $i - d(Classifier) \geq k$ then
            $X_{Classifier} = \{z_{d(Classifier)}, \dots, z_{d(Classifier)+k}\}$
            Retrain Model(Classifier) using $X_{Classifier}$ as training set
            $Modelisupdated(Classifier) = true$
            Update noncomformity function $A^{classifier}$ using the Model(Classifier)
            $S_i^{Classifier} = 1$
        end
    end
    The predicted label $pred_i$ for instance $z_i$ is calculated as the mode of the predicted
     labels from the 10 classifiers, i.e., the label that occurs most frequently in the set of
     predictions.
end
```

Algorithm 2: CD Detection using Multiple Classifiers

5.1. Datasets

5.1.1. Synthetic Benchmark Datasets

The STAGGER dataset (Schlimmer and Granger, 1986) is a standard benchmark dataset for evaluating CD detection. To conduct our simulations, we generated one million instances consisting of three categorical attributes and a binary target output. The drift type we considered is sudden, with four different concepts. A drift occurs every 10,000 examples, which corresponds to the chunk size.

The SEA (Street and Kim, 2001) dataset is a popular synthetic dataset that contains sudden CD. We have generated 1,000,000 instances for our simulations, where each example is described by 3 numeric attributes and has a binary label. There are four concepts and drift occurs every 250,000 examples (i.e. each chunk consists of 250,000 examples). In this dataset, the three variables take random values in the range [0-10] and if the sum of the first two variables is less than or equal to a pre-specified threshold, then the instance is assigned to class 1 otherwise to 0, the third variable is irrelevant. It is worth noting that here we test the transition from concept $a \rightarrow b \rightarrow c \rightarrow d$ while in the case of STAGGER, we test the transition from concept $a \rightarrow b \rightarrow c \rightarrow d \rightarrow a, \ldots$. It should be noted that in our study, we introduced noise to both datasets by changing 10% of the labels. We evaluated the performance of our approach with and without noise to assess its robustness to noisy data.

5.1.2. Real World Benchmark Datasets

The ELEC dataset (Harries et al., 1999) is a time series dataset containing 45,312 instances recorded at 30-minute intervals with a binary class label that indicates whether there has been a rise or a drop in price compared to a moving average of the last 24 hours. Each instance in the dataset consists of eight variables. The data has been collected from the Australian New South Wales Electricity Market. In our experiments, we excluded the variables related to time, date, day, and period and only used nswprice, nswdemand, transfer, vicprice, and vicdemand as predictors. We aimed to predict future values using a training set that consists of observations from less than one week. Note that sorting the data in this dataset based on the time it was received is crucial.

The Airlines dataset, referenced in (Ikonomovska E, 2010), is a collection of flight arrival and departure records of commercial flights within the USA from October 1987 to April 2008. The data was gathered from various USA airports and contains 539,383 instances, each described by seven features. The main objective of this dataset is to classify whether a given flight was delayed or not.

5.2. Results

In this section, we present the results of experiments conducted on the four benchmark datasets, where we employed an ICM ensemble composed of 10 treebaggers. Each treebagger contains 40 trees constructed using the bootstrap aggregating (bagging) technique. Specifically, subsets of the training data are randomly selected with replacement, and the resulting trees are combined using majority voting to form the final ensemble. For each example, the ten classifiers provide prediction labels, and the final prediction is obtained through majority voting. Each classifier outputs the posterior probability $\tilde{p}_j$ for the true label y_j, and the NCM is defined as $\alpha_j = -\tilde{p}_j$. As mentioned before, a separate sequence of p-values is computed for each classifier, and we apply ICM on each produced sequence of p-values to detect CD. Once a CD is detected, the current classifier ceases predictions and is retrained after a sufficient number of observations have been collected. We use $k = 100, 200, \ldots, 1000$ and $r = 2, 10, 100$ to determine the new training set for each classifier, as described in Algorithm 2. Specifically, we retrain the specific classifier when the Martingale value of the corresponding classifier exceeds a value $1/\delta$, where $\delta = 0.01$ is a significance level.

We use the Cautious betting function with parameters ϵ and W equal to 100 and $5,000$, respectively, for all experiments. The Cautious betting function is built on top of the histogram or the kernel density estimator. Our experiments use the p-values of the last $L = 1,000$ observations to calculate the histogram density estimator, as in (Eliades and Papadopoulos, 2022) with a number of bins equal to 15. When using the KDE, we set $L = 100$, as in (Volkhonskiy et al., 2017).

In the remaining two subsections, we perform simulations on the two Synthetic and two Real world datasets, focusing on improving the accuracy compared to our previous work. The presented results are averaged over five simulations on each dataset. Additionally, we compare the accuracy of the proposed approach with that of two state-of-the-art algorithms: AWE and DWM-NB, described in Section 2. The accuracies of these two algorithms were obtained from Sarnovsky and Kolarik (2021).

5.2.1. Synthetic Datasets

This subsection reports the ICM ensemble approach's experimental results on the STAGGER and SEA datasets.

The results of simulations on the noise-free and noise-injected STAGGER datasets are presented in Table 1. The findings indicate that our new approach with the histogram and Cautious betting function has slightly inferior performance to the Cautious betting function with the kernel. Nonetheless, our approach performs similarly to our previous study on this dataset. The limited improvement can be attributed to the ease of selecting a representative training set with ICM and only one classifier, particularly given that this dataset comprises only three categorical variables and a binary target output. Furthermore, the maximum theoretical accuracy is 1 with no noise in the dataset and 0.95 with 10% noise, leaving little room for improvement on this dataset. Here in most cases as r decreases we have a slight improvement in accuracy. Finally, the number of instances with no prediction in all scenarios remains below 1%.

Table 1: STAGGER

Betting function	**Noise**	**r**	**Accuracy**	**Instances with no prediction** (%)
Histogram with Cautious	0	100	0.995	0.79
	0	10	0.995	0.80
	0	2	0.996	0.79
	10	100	0.947	0.86
	10	10	0.948	0.84
	10	2	0.948	0.85
Kernel with Cautious	0	100	0.996	0.84
	0	10	0.997	0.86
	0	2	0.997	0.83
	10	100	0.949	0.91
	10	10	0.949	0.93
	10	2	0.949	0.91

The experiments on the SEA datasets presented in Table 2 reveal that for the noise-free dataset, the Cautious with Histogram approach outperforms the Cautious with Kernel approach. This is because the ICM ensemble algorithm with kernel fails to detect any concept drift. Therefore, it did not update the model, which was consistent with the previous study's findings (Eliades and Papadopoulos, 2022). However, both betting functions performed similarly when simulations were conducted on the noise-injected dataset. The proposed approach achieves slightly higher accuracy than the previous study. Nevertheless, the limited improvement can be attributed to the fact that using ICM with only one classifier makes it easy to select a representative training set without a more advanced strategy, particularly given that this dataset consists of only three numerical variables and binary target output. It's worth noting that, unlike the STAGGER dataset, decreasing the value of r in this dataset does not appear to improve accuracy. However, unlike the STAGGER dataset, decreasing the value of r in this dataset does not appear to improve accuracy. However, we only observe three changes here, as opposed to 99 changes in the STAGGER dataset, which makes it more challenging to determine the exact effect of r on our results. Additionally, the percentage of instances without prediction across all scenarios is less than 1%.

Table 2: SEA

Betting function	**Noise**	**r**	**Accuracy**	**Instances with no prediction** (%)
Histogram with Cautious	0	100	0.983	0.01
	0	10	0.982	0.01
	0	2	0.982	0.01
	10	100	0.917	0.01
	10	10	0.915	0.01
	10	2	0.916	0.01
Kernel with Cautious	0	100	0.922	0.01
	0	10	0.922	0.01
	0	2	0.922	0.01
	10	100	0.916	0.01
	10	10	0.916	0.01
	10	2	0.916	0.01

5.2.2. Real Datasets

In this subsection, we present the simulated results for the ELEC and the AIRLINES datasets while using the ICM ensemble.

Table 3 displays the outcomes of simulations performed on the ELEC dataset. The utilization of different betting functions yielded comparable accuracy outcomes. Nevertheless, comparing these accuracies with our previous study demonstrates an enhancement of roughly 1.7% by employing the Cautious histogram betting function and 0.7% when using the Cautious kernel betting function. The integration of classifiers with varying numbers of training instances seems to have contributed to improving accuracy. Additionally, our current study has predicted a substantially higher number of instances, with the number of instances with no prediction available being less than 1.7%. For most cases in this dataset, decreasing the value of r slightly improves the accuracy.

Table 3: ELEC

Betting function	**r**	**Accuracy**	**Instances with no prediction** (%)
Histogram with Cautious	100	0.765	0.41
	10	0.766	0.51
	2	0.767	0.39
Kernel with Cautious	100	0.764	1.33
	10	0.767	1.51
	2	0.766	1.63

The results of simulations on the AIRLINES dataset are presented in Table 4. The use of different betting functions led to similar accuracy results, with the Cautious kernel betting function performing slightly better than others. Comparing these accuracies with our previous study showed an improvement of around 4.5% when using the Cautious histogram betting function and 3.6% when using the Cautious kernel betting function. The use of classifiers with varying numbers of training instances appears to have contributed to improving accuracy. Additionally, our current study made predictions for a significantly larger number of instances, leading to higher accuracy compared to our earlier research (Eliades and Papadopoulos, 2022). Moreover, the percentage of instances with no available prediction remained below 1%. Again here for most cases in this dataset, decreasing the value of r slightly improves accuracy.

Table 4: AIRLINES

Betting function	**r**	**Accuracy**	**Instances with no prediction** (%)
Histogram with Cautious	100	0.635	0.02
	10	0.637	0.02
	2	0.637	0.02
Kernel with Cautious	100	0.638	0.02
	10	0.639	0.02
	2	0.640	0.02

5.3. Discussion

Table 5 compares the best betting function obtained from our simulations for each dataset with two state-of-the-art algorithms and the best outcome from our prior research (in third column), which utilized a single classifier. The table shows that our proposed approach achieves similar accuracy to the two state-of-the-art methods in all cases and outperforms both of them in the 10% noise SEA and 10% noise STAGGER dataset, as well as one of them in the AIRLINES dataset.

Additionally, the ICM ensemble either outperforms or achieves comparable results to the ICM with a single classifier (Eliades and Papadopoulos, 2022) in every scenario. Our study shows that the results obtained through simulations using the ICM ensemble method are superior to those reported by Eliades and Papadopoulos (2022). Notably, the number of available predictions is significantly increased, with no available predictions for at most 2% of the whole dataset. We conducted a z-test on the highest accuracy values obtained from the two methods. The proposed approach achieves significantly higher accuracy with a 1% significance level, except for the ELEC set, where we could not conclude any improvement even with a 5% significance level. It is worth noting that the z-test assumed an extreme scenario of perfect correlation between the accuracy values being compared and taking into account that each method has a different number of predictions available.

Table 5: Comparison of Cautious betting function with state of the art methods and our previous study

Dataset	**ICM ensemble**	**CAUTIOUS**	**AWE**	**DWE-NB**
STAGGER	0.949	0.946	0.948	0.901
SEA	0.917	0.915	0.879	0.876
ELEC	0.767	0.759	0.756	0.800
AIRLINES	0.640	0.602	0.618	0.640

6. Conclusions

We proposed an ensemble approach for change detection (CD) that has shown comparable and, in some cases, better accuracy than two state-of-the-art algorithms. The number of instances with no predictions is very low, specifically less than 2%. This is because the ensemble's performance is not affected when the independent ICM for a single model gives a false alarm, misses a change point, or is delayed in detecting a CD.

The proposed approach improves accuracy even when the ICM for a single classifier gives a false alarm. Applying ICM to a classifier's p-values may produce a false alarm, but the other classifiers can continue providing predictions, resulting in fewer instances with no available prediction. Moreover, if a classifier fails to detect or experiences a delayed alarm in the change point of the data-generating mechanism, the already updated classifiers will counterbalance the prediction, maintaining high accuracy. Additionally, the strategy of including training instances that arrived before the CD is detected has a positive impact on the classifiers' performance.

Our approach has great potential for addressing change detection, and there is room for further improvements. Therefore, our plans include investigating the combination of multiple classifiers and adopting strategies for selecting a representative training set for each underlying classifier.

References

Cesare Alippi and Manuel Roveri. Just-in-time adaptive classifiers—part i: Detecting nonstationary changes. *IEEE Transactions on Neural Networks*, 19(7):1145–1153, 2008. doi: 10.1109/TNN.2008.2000082.

Cesare Alippi, Giacomo Boracchi, and Manuel Roveri. Hierarchical change-detection tests. *IEEE Transactions on Neural Networks and Learning Systems*, 28(2):246–258, 2017. doi: 10.1109/TNNLS.2015.2512714.

Sikha Bagui and Katie Jin. A survey of challenges facing streaming data. *Transactions on Machine Learning and Artificial Intelligence*, 8(4):63–73, Aug. 2020. doi: 10.14738/tmlai.84.8579. URL https://journals.scholarpublishing.org/index.php/TMLAI/article/view/8579.

Li Bu, Dongbin Zhao, and Cesare Alippi. An incremental change detection test based on density difference estimation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(10):2714–2726, 2017. doi: 10.1109/TSMC.2017.2682502.

Charalambos Eliades and Harris Papadopoulos. Using inductive conformal martingales for addressing concept drift in data stream classification. In Lars Carlsson, Zhiyuan Luo, Giovanni Cherubin, and Khuong An Nguyen, editors, *Proceedings of the Tenth Symposium on Conformal and Probabilistic Prediction and Applications*, volume 152 of *Proceedings of Machine Learning Research*, pages 171–190. PMLR, 08–10 Sep 2021. URL https://proceedings.mlr.press/v152/eliades21a.html.

Charalambos Eliades and Harris Papadopoulos. A betting function for addressing concept drift with conformal martingales. In Ulf Johansson, Henrik Boström, Khuong An Nguyen, Zhiyuan Luo, and Lars Carlsson, editors, *Proceedings of the Eleventh Symposium on Conformal and Probabilistic Prediction with Applications*, volume 179 of *Proceedings of Machine Learning Research*, pages 219–238. PMLR, 24–26 Aug 2022. URL https://proceedings.mlr.press/v179/eliades22a.html.

Ryan Elwell and Robi Polikar. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531, 2011. doi: 10.1109/TNN.2011.2160459.

Valentina Fedorova, Alex Gammerman, Ilia Nouretdinov, and Vladimir Vovk. Plug-in martingales for testing exchangeability on-line. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 923–930, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

Ricky Fok, Aijun An, and Xiaogang Wang. Mining evolving data streams with particle filters. *Computational Intelligence*, 33(2):147–180, 2017. doi: https://doi.org/10.1111/coin.12071. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/coin.12071.

Michael Harries, U Nsw cse tr, and New South Wales. Splice-2 comparative evaluation: Electricity pricing. Technical report, 1999.

Shen-Shyang Ho. A martingale framework for concept change detection in time-varying data streams. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML 05, page 321–327, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102392. URL https://doi.org/10.1145/1102351.1102392.

Shen-Shyang Ho, Matthew Schofield, Bo Sun, Jason Snouffer, and Jean Kirschner. A martingale-based approach for flight behavior anomaly detection. In *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, pages 43–52, 2019. doi: 10.1109/MDM.2019.00-75.

Dveroski S Ikonomovska E, Gama J. Learning model trees from evolving data streams. *Data Mining and Knowledge Discovery*, 23:128–168, 2010.

J. Zico Kolter and Marcus A. Maloof. Dynamic weighted majority: An ensemble method for drifting concepts. *J. Mach. Learn. Res.*, 8:2755–2790, December 2007. ISSN 1532-4435.

Peipei Li, Xindong Wu, Xuegang Hu, and Hao Wang. Learning concept-drifting data streams with random ensemble decision trees. *Neurocomputing*, 166:68–83, 2015. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2015.04.024. URL https://www.sciencedirect.com/science/article/pii/S0925231215004713.

Jian-Wei Liao and Bi-Ru Dai. An ensemble learning approach for concept drift. In *2014 International Conference on Information Science & Applications (ICISA)*, pages 1–4, 2014. doi: 10.1109/ICISA.2014.6847357.

Anjin Liu, Yiliao Song, Guangquan Zhang, and Jie Lu. Regional concept drift detection and density synchronized drift adaptation. pages 2280–2286, 08 2017. doi: 10.24963/ijcai.2017/317.

Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12): 2346–2363, 2019. doi: 10.1109/TKDE.2018.2876857.

Martin Sarnovsky and Michal Kolarik. Classification of the drifting data streams using heterogeneous diversified dynamic class-weighted ensemble. *PeerJ Computer Science*, 7, 04 2021. doi: 10.7717/peerj-cs.459.

Mark J. Schervish. *Theory of Statistics*. Springer, New York, 1995.

Jeffrey C. Schlimmer and Richard H. Granger. Incremental learning from noisy data. *Mach. Learn.*, 1(3):317–354, March 1986. ISSN 0885-6125. doi: 10.1023/A:1022810614389. URL https://doi.org/10.1023/A:1022810614389.

Bernard W Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, London, 1986. URL https://cds.cern.ch/record/1070306.

W. Nick Street and YongSeog Kim. A streaming ensemble algorithm (sea) for large-scale classification. In *Proceedings of the Seventh ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining, KDD '01, page 377–382, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 158113391X. doi: 10.1145/502512.502568. URL https://doi.org/10.1145/502512.502568.

J. Ville. Étude critique de la notion de collectif. by j. ville. pp. 144. 75 francs. 1939. monographies des probabilités, calcul des probabilités et ses applications, publiées sous la direction de m. Émile borel, fascicule iii. (gauthier-villars, paris). *The Mathematical Gazette*, 23(257):490–491, 1939. doi: 10.2307/3607027.

Denis Volkhonskiy, Evgeny Burnaev, Ilia Nouretdinov, Alexander Gammerman, and Vladimir Vovk. Inductive conformal martingales for change-point detection. In Alex Gammerman, Vladimir Vovk, Zhiyuan Luo, and Harris Papadopoulos, editors, *Proceedings of the Sixth Workshop on Conformal and Probabilistic Prediction and Applications*, volume 60 of *Proceedings of Machine Learning Research*, pages 132–153, Stockholm, Sweden, 13–16 Jun 2017. PMLR. URL http://proceedings.mlr.press/v60/volkhonskiy17a.html.

Vladimir Vovk. Testing for concept shift online, 2020.

Vladimir Vovk, Ilia Nouretdinov, and Alexander Gammerman. Testing exchangeability on-line. pages 768–775, 01 2003.

Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, page 226–235, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581137370. doi: 10.1145/956750.956778. URL https://doi.org/10.1145/956750.956778.