

Online Boosting Adaptive Learning under Concept Drift for Multistream Classification

En Yu, Jie Lu*, Bin Zhang, Guangquan Zhang

Decision Systems and e-Service Intelligence Laboratory, Australian Artificial Intelligence Institute (AAIL),
Faculty of Engineering and Information Technology, University of Technology Sydney, Australia
En.Yu@student.uts.edu.au; {Jie.Lu, Bin.Zhang, Guangquan.Zhang}@uts.edu.au

Abstract

Multistream classification poses significant challenges due to the necessity for rapid adaptation in dynamic streaming processes with concept drift. Despite the growing research outcomes in this area, there has been a notable oversight regarding the temporal dynamic relationships between these streams, leading to the issue of negative transfer arising from irrelevant data. In this paper, we propose a novel **Online Boosting Adaptive Learning (OBAL)** method that effectively addresses this limitation by adaptively learning the dynamic correlation among different streams. Specifically, OBAL operates in a dual-phase mechanism, in the first of which we design an **Adaptive COvariate Shift Adaptation (AdaCOSA)** algorithm to construct an initialized ensemble model using archived data from various source streams, thus mitigating the covariate shift while learning the dynamic correlations via an adaptive re-weighting strategy. During the online process, we employ a Gaussian Mixture Model-based weighting mechanism, which is seamlessly integrated with the acquired correlations via AdaCOSA to effectively handle asynchronous drift. This approach significantly improves the predictive performance and stability of the target stream. We conduct comprehensive experiments on several synthetic and real-world data streams, encompassing various drifting scenarios and types. The results clearly demonstrate that OBAL achieves remarkable advancements in addressing multistream classification problems by effectively leveraging positive knowledge derived from multiple sources.

Introduction

In various real-world scenarios, such as auto-driving systems, weather forecasts, and industrial production, data is continuously and sequentially generated over time, which is referred to as data streams or streaming data (Lu et al. 2018; Zhou et al. 2023b; Wang et al. 2022a). These data streams are susceptible to changes in their underlying distribution, resulting in concept drift. Consequently, classifiers trained on historical data may fail to predict subsequent samples, leading to a performance decrease (Li et al. 2022; Xu et al. 2023). Thus, it attracts many researchers to develop efficient learning techniques capable of analyzing streaming data with concept drift in non-stationary environments.

To date, prior studies have provided empirical evidence of the efficacy of concept drift adaptation methods in effectively addressing data streams with dynamic distributions. It is worth noting that the majority of existing techniques have been tailored specifically for a single stream with delayed labels (Yu et al. 2022b; Song et al. 2021b). However, it is common to encounter scenarios where multiple data streams are generated simultaneously in real-world intelligent systems. For example, data samples continuously stream from sensors in manufacturing systems. These data streams, despite being associated with the same task, often exhibit distinct distributions due to varying data sources (Zhou et al. 2023a). In addition, while data collection is straightforward, the labeling process incurs high time and labor costs, leading to the hybrid multiple streams where massive labeled and unlabeled streams arrive simultaneously (Yu et al. 2022a).

To tackle this scenario, multistream classification has been proposed, in which a model can be flexibly transferred from labeled source streams to the unlabeled target stream while employing online detection and adaptation working principles. This not only enables the model to adapt to new and unlabeled data streams but also mitigates the expenses and logistical challenges. The multistream classification problem features three major challenges that have to be tackled simultaneously: 1) *Scarcity of labels*: this arises from the absence of labels specifically for the target stream, while the source streams possess labeled data; 2) *Covariate shift*: this implies that any two data streams exhibit distinct distributions, whether they are different source streams or a source stream and a target stream; and 3) *Asynchronous drift*: the source and target streams are susceptible to independent concept drift, which occurs at varying time periods and results in unique effects on the model performance.

In recent years, several approaches have been proposed to address the multistream classification problem by using online domain adaptation and drift handling techniques (Chandra et al. 2016; Haque et al. 2017; Pratama et al. 2019; Wang et al. 2021). However, many of these methods have primarily focused on the single-source stream, potentially impeding model performance due to limitations in the quality of the source data. Furthermore, such single-source-based approaches may be prone to overfitting issues. Accordingly, the multi-source configuration is introduced, which enables the acquisition of supplementary information from different

*Corresponding Author

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

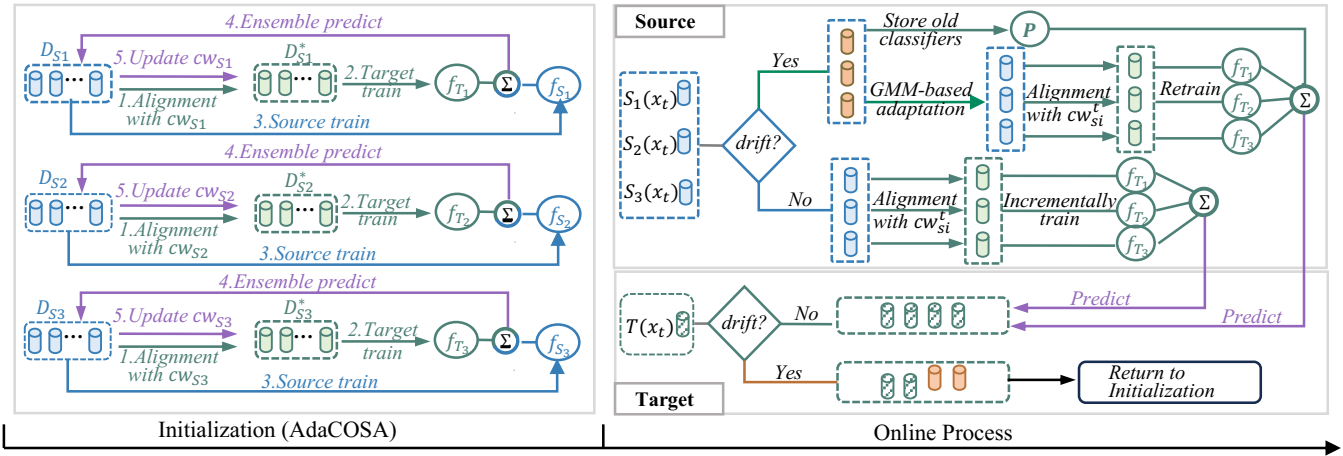


Figure 1: Framework of OBAL. The initialization stage is principally devoted to mitigating the problem of covariate shift, along with learning the intricate dynamic correlations that exist between various data streams. In the online phase, the core focus is on the detection and adaptation of asynchronous drift. This stage further integrates the covariate shift alignment and correlation matrices learned during the initial phase, facilitating a seamless ensemble prediction from the source to the target stream.

source streams, thereby providing more valuable information to build a more accurate and robust model (Wang et al. 2022b; Yang et al. 2021). However, leveraging the information from each individual source stream exposes a new challenge: 4) *Temporal dynamic correlations* between the source and target streams. In other words, any drift occurring within each stream has the potential to alter the correlation between the source and target streams. It is crucial for the predictive model to adapt promptly, extracting valuable insights from relevant source streams while avoiding the assimilation of irrelevant knowledge from other source streams.

To address all issues in the multi-stream classification task, we propose the **Online Boosting Adaptive Learning (OBAL)** method. As shown in Figure 1, OBAL consists of two stages, the first of which is the initialization phase, where we propose the AdaCOSA algorithm. The fundamental principle of AdaCOSA involves an adaptive interaction between models learned in the original source space and those acquired in the target space, aiming to align the *temporal covariate shift* and explore the *dynamic relationships* between different data streams based on feedback from the target domain. This process reinforces positive knowledge transfer, leading to optimal model migration. The second stage involves the online processing phase, during which our primary aim is to detect and adapt to the *asynchronous drift* in each data stream in real-time. To achieve this, we employ the Drift Detection Method (DDM) (Gama et al. 2004) for labeled source streams, as it offers a stable and accurate detection approach. Simultaneously, we utilize the Gaussian Mixture Model (GMM) (Oliveira, Minku, and Oliveira 2021) based weighting strategy for asynchronous drift adaptation in these streams. For the unlabeled target stream, we design two sliding windows and continuously monitor their distribution changes to effectively detect drift occurrences. Once a drift is detected in the target stream, it signifies that the dynamic relationships learned in the first stage are no

longer applicable, necessitating a return to the first stage for reinitialization. The main contributions of our work can be summarized as follows:

- This paper presents a new online ensemble approach (OBAL) for multi-source data stream classification. With the capability to dynamically detect and adapt to concept drift, OBAL demonstrates enhanced effectiveness and stability. Moreover, it offers effortless extensibility in managing diverse data streams.
- A novel algorithm (AdaCOSA) is proposed to align the covariate shift as well as investigate a new dynamic correlation issue between source and target streams. It further enhances positive knowledge transfer and prevents negative transfer effects.
- We design a simple yet effective GMM-based module to adapt the asynchronous drift. It orchestrates an ensemble of both historical classifiers and newly trained classifiers on weighted source samples. By accumulating abundant source knowledge, the proposed approach achieves improved prediction accuracy for the target stream.

Related Works

Date stream classification has become an increasingly critical area of research due to the dynamic nature of real-world data streams, i.e., concept drift. Concept drift refers to the underlying data distribution changing over time, which occurs as time $t + 1$ if joint distribution $P_{t+1}(X, y) \neq P_t(X, y)$. It poses significant challenges for classifiers to maintain accuracy and adapt promptly. To tackle the concept drift problem, many works have been proposed to ensure the effectiveness and reliability of models (Gomes et al. 2017; Miyaguchi and Kajino 2019; Chiu and Minku 2020; Jothimurugesan et al. 2023). However, most methods are designed for *single-labeled stream*, which is not suitable for the multi-stream scenario. To fill this research blank, Chan-

dra et al. (Chandra et al. 2016) introduce a multi-stream classification framework that utilizes ensemble classifiers for each data stream and incorporates Kernel Mean Matching to reduce the disparity between source and target streams. They further propose the FUSION algorithm (Haque et al. 2017) to leverage the Kullback Leibler Importance Estimation Procedure for density ratio estimation and covariate shift handling. In addition, some neural-network-based models are proposed to deal with high-dimensional data (Yoon et al. 2022). For example, Autonomous Transfer Learning (ATL) (Pratama et al. 2019) is an online domain adaptation strategy that employs both generative and discriminative phases, combined with Kullback Leibler divergence-based optimization. Moreover, Yu et al. (Yu et al. 2022b) propose a meta-learning-based framework to learn the invariant features of drifting data streams and then update the meta model in an online fashion.

In addition, multi-source stream classification is proposed to enhance the robustness by considering the complementary information from different source streams simultaneously. For example, Du et al. (Du, Minku, and Zhou 2019) introduced Melanie, which employs a weighted ensemble classifier to transfer knowledge from multiple source streams. It is the first approach capable of simultaneously transferring knowledge from various source streams with concept drift. However, Melanie is a supervised method, which cannot be used for unlabeled data prediction.

Hence, the AutOMatic Multi-Source Domain Adaptation (AOMSDA) (Renchunzi and Pratama 2022) incorporates a central moment discrepancy-based regularizer to leverage the complementary information from multi-source streams, and employs a node weighting strategy to tackle the covariate shift. AOMSDA is a chunk-based method, which means it lacks the ability to dynamically detect the changes in data streams. To address this limitation, Jiao et al. (Jiao et al. 2022) propose a reduced-space Multistream Classification based on Multi-objective Optimization (MCMO). It seeks a common feature subset to minimize the distribution shift and then uses a GMM to detect and adapt asynchronous drift. However, all these methods determine the correlation between each individual source and target stream as fixed, which does not fully exploit temporal dynamic correlations.

Proposed Method

Problem Definition

Multi-source-stream classification involves the presence of multiple labeled source streams and one unlabeled target stream. These streams possess interconnected internal representations and share a common label space. The objective of this task is to predict the labels of the target stream by effectively transferring knowledge from the labeled source to the target stream, and it can be defined as follows.

Definition 1 Multi-source-stream Classification. It involves N labeled source streams $S = \{S_1, S_2, \dots, S_N\}$ and one unlabeled target stream T . Each arrived data sample at time t is represented by $S_i(\mathbf{x}_t, y_t)$, where $\mathbf{x}_t \in \mathcal{D}^d$ is the d -dimensional features, and y_t is the true label of the instance which can only be observed in $S_i, i \in \{1, 2, \dots, N\}$.

Algorithm 1: Initialization (AdaCOSA)

Input: The archived data batches $D_{S_i}, i \in \{1, 2, \dots, N\}$ and D_T , Maximum iteration I_{max} .
Output: target classifier f_{T_i} , weight vector \mathbf{cw}_{S_i} .
1: Get the mapped source data $D_{S_i}^*$ according to Eq.3.
2: Set up β_n and initialize \mathbf{cw}_{S_i} .
3: **for** $iter = 1 : I_{max}$ **do**
4: **for** $i = 1 : N$ **do**
5: Create source classifiers $f_{S_i}(x) \leftarrow \{D_{S_i}, Y\}$.
6: Create target classifiers $f_{T_i}(x) \leftarrow \{D_{S_i}^*, Y, \mathbf{cw}_{S_i}\}$
7: Predict the instance from D_{S_i} using F_{est} .
8: Adjust the weight vector \mathbf{cw}_{S_i} according to Eq.4.
9: **end for**
10: **end for**

It aims to build a classification model to predict the class label of $T(\mathbf{x}_t)$ using the $S_i(\mathbf{x}_t, y_t)$ and $T(\mathbf{x}_t)$.

As mentioned before, four main challenges must be addressed simultaneously in the multistream classification problem, i.e., *scarcity of labels, covariate shift, asynchronous drift* and *dynamic correlation*. These challenges are defined as follows,

Challenge 1 Scarcity of Labels. *This is a major issue in the multistream classification problem. Labeled samples are provided only to the source streams $S_i(\mathbf{x}_t, y_t), i \in \{1, 2, \dots, N\}$, leaving the target stream entirely unlabelled $T(\mathbf{x}_t)$. Consequently, the challenge lies in achieving accurate predictions in the target stream, where no labeled samples are available.*

Challenge 2 Covariate Shift. *Denoting P_{S_i} and P_T as the distributions from $S_i, i \in 1, 2, \dots, N$ and T , all streams at the same time step are related but with covariate shift, i.e., $P_{S_i}(y_t | \mathbf{x}_t) = P_{S_j}(y_t | \mathbf{x}_t) = P_T(y_t | \mathbf{x}_t)$ while $P_{S_i}(\mathbf{x}_t) \neq P_{S_j}(\mathbf{x}_t) \neq P_T(\mathbf{x}_t)$*

Challenge 3 Asynchronous Drift. *This refers to the observation of the effect of drift at different times on different independent non-stationary processes that continuously generate data from $S = \{S_1, S_2, \dots, S_N\}$ and T .*

- **Source Drift:** $\exists t$ if $P_{S_i}(\mathbf{x}_t) \neq P_{S_i}(\mathbf{x}_{t+1}), i \in 1, 2, \dots, N$ but $P_T(\mathbf{x}_t) = P_T(\mathbf{x}_{t+1})$, the drift only occurs in the source stream.
- **Target Drift:** $\exists t$ if $P_{S_i}(\mathbf{x}_t) = P_{S_i}(\mathbf{x}_{t+1}), i \in 1, 2, \dots, N$ but $P_T(\mathbf{x}_t) \neq P_T(\mathbf{x}_{t+1})$, the drift only occurs in the target stream.
- **Concurrent Drifts:** $\exists t$ if $P_{S_i}(\mathbf{x}_t) \neq P_{S_i}(\mathbf{x}_{t+1}), i \in 1, 2, \dots, N$ and $P_T(\mathbf{x}_t) \neq P_T(\mathbf{x}_{t+1})$, it means drift occurs in both source and target streams.

Challenge 4 Temporal Dynamic Correlation. *The dynamic interplay between source and target streams leads to varying relevance, expressed as $C(S_i(\mathbf{x}_t), T(\mathbf{x}_t))$. At the time t , some source streams may possess complementary information $C(S_i(\mathbf{x}_t), T(\mathbf{x}_t)) = +$, while others may contain negative information $C(S_i(\mathbf{x}_t), T(\mathbf{x}_t)) = -$. The complexity arises as C may change over time, such as*

$C(S_i(\mathbf{x}_t), T(\mathbf{x}_t)) \neq C(S_i(\mathbf{x}_{t+\tau}), T(\mathbf{x}_{t+\tau}))$, disrupting the inherent relationship between the streams.

To address all challenges, we propose the OBAL method which comprises two stages: initialization (AdaCOSA) and online processing. Next, we will provide a detailed description of these two stages.

Adaptive Covariate Shift Adaptation (AdaCOSA)

To align the covariate shift $P_{S_i}(\mathbf{x}_t) \neq P_{S_j}(\mathbf{x}_t) \neq P_T(\mathbf{x}_t)$ as well as to explore the temporal dynamic relationship $C(S_i(\mathbf{x}_t), T(\mathbf{x}_t))$ between source and target streams, we propose an AdaCOSA algorithm. Inspired by the CORrelation ALignment (CORAL) method (Sun, Feng, and Saenko 2016), the covariance between shifting domains can be aligned by minimizing the distance between the second-order statistics, which provides a stable and effective solution. However, the standard CORAL method is incapable of identifying source instances that are irrelevant to the target, thereby leading to negative transfer effects (Wang et al. 2019; Yang et al. 2021). Furthermore, it fails to address the dynamic relationship between the data streams. As a solution, we propose an adaptive re-weighting strategy to dynamically and iteratively adjust the weights of the source data based on their relevance to the target domain.

Specifically, given any archived source data batch $D_{S_i} = S_i(\mathbf{X}, Y)$, $i \in \{1, 2, \dots, N\}$ and the target data batch $D_T = T(\mathbf{X})$, we first assign a correlation weight vector $\mathbf{c}w_{S_i} = [cw_{S_i}^1, cw_{S_i}^2, \dots, cw_{S_i}^{L_n}]$, $i \in \{1, 2, \dots, N\}$ to each source stream, where L_n is the instance number of each archived data batch. Then we can align the shifting covariance by mapping each weighted source data to the target domain using a transformation matrix A_{S_i} , and the objective can be formulated as,

$$\min_{A_{S_i}} \left\| C_{S_i} - C_T \right\|_F^2 = \min_{A_{S_i}} \left\| A_{S_i}^\top C_{S_i} A_{S_i} - C_T \right\|_F^2, \quad (1)$$

where $\|\cdot\|_F$ is the Frobenius norm. C_{S_i} and C_T are the covariance matrices of $\mathbf{c}w_{S_i} D_{S_i}$ and D_T , respectively. C_{S_i} is the covariance matrix of transformed source features $\mathbf{c}w_{S_i} D_{S_i} A$, and

$$\begin{aligned} C_{S_i} &= \text{cov}(\mathbf{c}w_{S_i} D_{S_i}) + \text{eye}(\text{size}(\mathbf{c}w_{S_i} D_{S_i}, 2)), \\ C_T &= \text{cov}(D_T) + \text{eye}(\text{size}(D_T, 2)). \end{aligned} \quad (2)$$

Then the aligned source data $D_{S_i}^*$ can be obtained by the classical whitening and re-coloring strategy (Sun, Feng, and Saenko 2016) (Please refer to Supplementary S1 for the detailed theoretical analysis),

$$D_{S_i}^* = \mathbf{c}w_{S_i} D_{S_i} C_{S_i}^{-\frac{1}{2}} C_T^{\frac{1}{2}}. \quad (3)$$

Next, we use a supervised method to train the source classifiers f_{S_i} using raw source data $\{D_{S_i}, Y\}$. In addition, the covariate-adopted target classifiers f_{T_i} can be learned by using the transformed $\{D_{S_i}^*, Y\}$. Finally, we can employ an average ensemble F_{est} that combines models derived from each original source space f_{S_i} with those learned in the target space f_{T_i} to re-evaluate the source data iteratively.

Once the predicted label \hat{y}_i is obtained, it can be used to re-estimate the correlation weights of the source instances

Algorithm 2: The learning process of OBAL

Input: Source streams $\{S_1, S_2, \dots, S_N\}$, target stream T , classifier pool P , initial sample size L_n .

Output: Predicted labels for target stream.

```

1:  $D_{S_i}, D_T \leftarrow$  Read first  $L_n$  instances from  $S_i$  and  $T$ .
2:  $f_E(x), \mathbf{c}w_{S_i} \leftarrow$  initialize according to Algorithm 1.
3: Create  $DDM_{S_i}$  and  $GMM_{S_i}$  for source stream.
4: Create  $GMM_T$  for target stream.
5: Create detection and reference windows  $W_{det}, W_{ref}$ .
6: while there is incoming data do
7:   for  $i = 1 : N$  do
8:     if  $DDM_{S_i} = \text{True}$  then
9:       GMM-based adaptation by Eq.8.
10:      Weighted alignment and retrain a new classifier.
11:     else
12:       Weighted alignment and incrementally update.
13:     end if
14:   end for
15:   Move detection window and calculate  $\mu_{det}, \mu_{ref}$ .
16:   if Eq. 11 = True then
17:     Remove all base classifiers and return to line 1.
18:   else
19:     Predict the target sample.
20:   end if
21: end while

```

because it contains reliable responses from the target domain. In each iteration, if the source instance is predicted mistakenly, this instance may likely conflict with the target stream. Then the effect of this irrelevant data will be diminished in the next iteration by decreasing its training weight. In contrast, accurate predictions indicate a minimal distance or positive correlation between the source and target domains, resulting in increased training weights to enhance learning. Here, the weight can be updated by,

$$cw_{S_i}^t = cw_{S_i}^t \cdot e^{-\beta_n |\hat{y}_i - y_i|}, \quad (4)$$

where β_n is a hyper-parameter defined as $\beta_n = 0.5 \ln \left(1 + \sqrt{2 \ln \frac{L_n}{I_{\max}}} \right)$. L_n is the total number of samples of the archived data batch D_{S_i} , and I_{\max} is the maximum iterations for adaptive re-weighting.

After several iterations, the instances that exhibit a positive correlation with the target stream will be assigned higher training weights, whereas the training instances that diverge from the target stream will receive lower weights. The detailed process is presented in Algorithm 1. After that, the weight cw_{S_i} of each target base classifier can be assigned based on the learned correlation weight and it is calculated by $cw_{S_i} = \frac{1}{L_n} \sum_{t=1}^{L_n} cw_{S_i}^t$. Therefore, the final ensemble f_E for the target stream can be formulated as follows:

$$f_E(x) = \frac{cw_{S_i}}{\sum_{i=1}^N cw_{S_i}} f_{T_i}. \quad (5)$$

Online Detection and Adaptation

As stated in Challenge 3, asynchronous concept drifts may occur in either the source or target streams over time. Therefore, for any given stream, it is necessary to continuously

	SEA	Tree	RBF	Hyperplane	Weather	Kitti	CNNIBN	BBC
FUSIONs1	85.04 \pm 0.84	76.98 \pm 1.11	82.03 \pm 1.41	83.29 \pm 0.67	71.04 \pm 1.50	54.21 \pm 2.61	66.76 \pm 0.74	61.76 \pm 0.09
FUSIONs2	85.78 \pm 0.92	76.74 \pm 1.00	83.46 \pm 1.20	84.05 \pm 0.52	70.65 \pm 1.32	52.36 \pm 2.72	67.54 \pm 1.11	61.26 \pm 0.43
FUSIONs3	84.31 \pm 1.13	75.21 \pm 1.07	81.03 \pm 1.73	82.17 \pm 0.57	72.17 \pm 1.17	50.38 \pm 2.43	65.34 \pm 0.92	59.86 \pm 0.19
ATLs1	88.42 \pm 1.70	76.43 \pm 2.17	84.53 \pm 2.01	86.17 \pm 1.04	74.57 \pm 1.94	52.78 \pm 3.78	62.78 \pm 1.44	62.78 \pm 1.16
ATLs2	88.74 \pm 1.75	76.71 \pm 1.86	85.21 \pm 1.85	87.07 \pm 1.21	75.03 \pm 2.01	54.01 \pm 3.09	65.74 \pm 1.76	62.34 \pm 0.83
ATLs3	87.62 \pm 1.01	76.07 \pm 2.42	83.16 \pm 2.13	86.01 \pm 1.49	74.62 \pm 1.77	53.26 \pm 3.21	62.65 \pm 1.38	60.76 \pm 0.77
Melanie	89.18 \pm 0.77	78.93 \pm 0.61	86.04 \pm 0.39	86.38 \pm 0.57	77.74 \pm 0.89	50.29 \pm 1.34	68.79 \pm 0.31	68.04 \pm 0.01
AOMSDA	90.23 \pm 1.42	76.87 \pm 3.47	85.26 \pm 2.89	87.66 \pm 1.74	76.55 \pm 1.41	67.79 \pm 3.16	69.07 \pm 1.40	63.36 \pm 1.07
MCMO	87.46 \pm 2.12	77.64 \pm 1.47	86.26 \pm 0.77	84.04 \pm 1.42	76.02 \pm 3.43	64.82 \pm 4.17	68.83 \pm 0.89	60.12 \pm 1.51
OBAL (ours)	90.98 \pm 0.87	78.45 \pm 1.01	86.78 \pm 0.91	88.01 \pm 1.17	79.22 \pm 2.07	70.29 \pm 3.42	70.71 \pm 0.77	66.43 \pm 1.42

Table 1: Classification accuracy (%) with the variance of various methods on all benchmarks.

monitor its drifting situation in real-time and promptly perform drift adaptation to accommodate the new concept.

Source Stream Processing. For scenarios involving source drift, existing supervised drift detectors such as DDM can be employed, which offers more accurate drift detection because of the leveraging of labels. As a new source sample $S_i(\mathbf{x}_t)$ arrives, the source classifier predicts its label, and then the drift detector is updated based on the prediction error. If no drift is detected, we will incrementally train the target classifier using the weighted mapped $S_i^*(\mathbf{x}_t^*, y)$ with its corresponding weight $cw_{S_i}^t$. Since we have obtained the optimal weights $\text{cw}_{S_i} = [cw_{S_i}^1, cw_{S_i}^2, \dots, cw_{S_i}^L]$ for the archived data batch during the initialization stage, we can retrieve the most relevant data from the archived data batch and assign its weights to the new coming data by indexing the minimum L_2 distance between new coming and archived data instances. However, once a drift is detected within each source stream, an adaptation module should be deployed to handle new concepts. Here, we utilize the GMM to evaluate the distributions of the old and new concepts. GMM assumes several mixture components can model all real-world data, and it is formulated as follows:

$$P(x) = \sum_{k=1}^K P(x | C_k) \cdot w_k, \quad (6)$$

where K represents the total number of Gaussians or mixture components, and x is the observed multivariate. w_k is a weight that is determined by the observations that constitute C_k , and $0 \leq w_k \leq 1$, $\sum_{k=1}^K w_k = 1$. $P(x | C_k)$ represents the likelihood of observation x being assigned to mixture component C_k . It can be calculated by using the mean μ_k and the covariance Σ_k of each mixture component C_k :

$$P(x | C_k) = \frac{1}{\sqrt{2\pi^{d/2} |\Sigma_k|}} \exp\left(-\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\right). \quad (7)$$

According to the Expectation-Maximization (EM) algorithm, all the parameters of different mixture components are randomly initialized using the archived data batch D_{si} . Subsequently, it iteratively adjusts the mean and covariance of the mixture component to maximize the likelihood of each mixture component. For a newly incoming instance

$S_i(\mathbf{x}_t)$, its importance weight $aw_{S_i}^t$ can be calculated by maximizing the conditional probability of GMM as follows:

$$aw_{S_i}^t = \max_{k \in \{1, 2, \dots, K\}} P(S_i(\mathbf{x}_t) | C_k). \quad (8)$$

Then, the new coming concept in any source stream can be adapted to the old concept by multiplying $aw_{S_i}^t$. Thus, its optimal correlation weight $cw_{S_i}^t$ with the target stream can also be obtained from the learned cw_{S_i} . Finally, a new target base classifier will be created and trained by using weighted mapped $S_i^*(\mathbf{x}_t^*, y)$ with its corresponding weight $cw_{S_i}^t$. Note that old base classifiers are no longer trained with new samples but are instead preserved within a base classifier pool denoted as P , allowing for their retention. Finally, the joint predictive probability can be ensembled as,

$$f^E(x) = \frac{w_{S_i}}{\sum_{i=1}^N w_{S_i} + \sum_{l=1}^{|P|} w_P} f_{T_i}(x) + \frac{w_P}{\sum_{i=1}^N w_{S_i} + \sum_{l=1}^{|P|} w_P} f_P(x), \quad (9)$$

where w_P is the weight of l -th classifier in P , and $w_{S_i} = \frac{1}{n} \sum_{t=1}^n aw_{S_i}^t * cw_{S_i}^t$.

Target Stream Processing. To detect the drift in the target stream without utilizing labels, we use the archived target data batch D_T to initialize a GMM model and deploy two sliding windows to detect the changes over time. Specifically, we design two sliding windows, i.e., Reference Window $W_{ref} = \{T(x_1), \dots, T(x_n)\}$ and Detect Window $W_{det} = \{T(x_{n+1}), \dots, T(x_{2n})\}$, where n is the instance number within the window and it is set as $n = L_n$. Then, the average conditional probability of the reference window can be calculated by a point estimation of the mean for the normal distribution,

$$\mu_{ref} = \frac{1}{n} \sum_{t=1}^n \max_{k \in \{1, 2, \dots, K\}} P(T(\mathbf{x}_t) | C_k). \quad (10)$$

The confidence interval estimation of the μ_{ref} is known to be $[\mu_{ref} - z_\alpha(\sigma/\sqrt{n}), \mu_{ref} + z_\alpha(\sigma/\sqrt{n})]$, where σ is the standard deviation and z_α is the significance level which is set as 3 (Kim and Park 2017). The decision is made that the change has occurred when the point estimation by the mean μ_{ref} in the detection window satisfies,

$$\mu_{det} \geq \mu_{ref} + z_\alpha \times \sigma / \sqrt{n}. \quad (11)$$

Otherwise, W_{ref} and W_{det} move step by step to receive new incoming data, i.e., $W_{ref} = \{T(x_t), \dots, T(x_{n+t-1})\}$ and $W_{det} = \{T(x_{n+t}), \dots, T(x_{2n+t-1})\}$. Once a change is detected, the historical base classifier becomes ineffective for classifying target samples. Consequently, all base classifiers are eliminated from the classifier pool, and the model undergoes re-initialization to adapt to the new concepts. The learning process is summarized in Algorithm 2.

Experiments

In the experiment, we first empirically demonstrated that OBAL consistently outperforms current methods, highlighting both robustness and superiority. Second, we validated the substantial impact of dynamic inter-stream relationships on prediction, emphasizing the effectiveness of the AdaCOSA by ablation study. Additionally, we confirmed OBAL’s scalability across various data streams, validating its consistent predictive performance. Finally, we assessed parameter sensitivity, time complexity, and algorithmic cost.

Experiment Settings

Benchmark Datasets. We conduct the experiment on four synthetic datasets (i.e., SEA (Street and Kim 2001), Tree (Liu, Lu, and Zhang 2020), RBF (Song et al. 2021a), and Hyperplane (Bifet and Gavalda 2007)) and four popular real-world datasets (Weather (Ditzler and Polikar 2012), Kitti (Geiger, Lenz, and Urtasun 2012), CNNIBN (Vyas et al. 2014), and BBC (Vyas et al. 2014)), and more detailed descriptions of each dataset and multistream scenario simulation can be found in Supplementary S3 and Table S1.

Baselines. To demonstrate the superiority of our proposed method, we conducted experiments comparing it with five state-of-the-art methods. Among them, the FUSION (Haque et al. 2017) and ATL (Pratama et al. 2019) algorithms are based on single-source streams, while the Melanie (Du, Minku, and Zhou 2019), AOMSDA (Renchunzi and Pratama 2022), and MCMO (Jiao et al. 2022) are specifically designed for the multi-source classification scenario. For FUSION and ATL, we pair each source stream with the target stream, resulting in three distinct groups denoted as $\{\text{FUSIONs1}, \text{FUSIONs2}, \text{and FUSIONs3}\}$ and $\{\text{ATLs1}, \text{ATLs2}, \text{ATLs3}\}$, respectively.

Results Analysis

Overall Performance. Table 1 compares the classification accuracy of OBAL against all baselines on four synthetic and four real-world datasets. Overall, OBAL outperforms all other unsupervised multistream classification methods on both synthetic and real-world datasets, while it performs better than the supervised method (Melanie) on six out of eight datasets. First, compared to single-source-based methods (Fusion and ATL), all multi-source-based methods demonstrate significant improvement. This proves that multiple labeled source streams can provide more discriminative and complementary information, resulting in more accurate and robust predictions. Compared with Melanie, OBAL performs remarkably close to or even surpasses without considering the target labels. This is because we not only mitigate

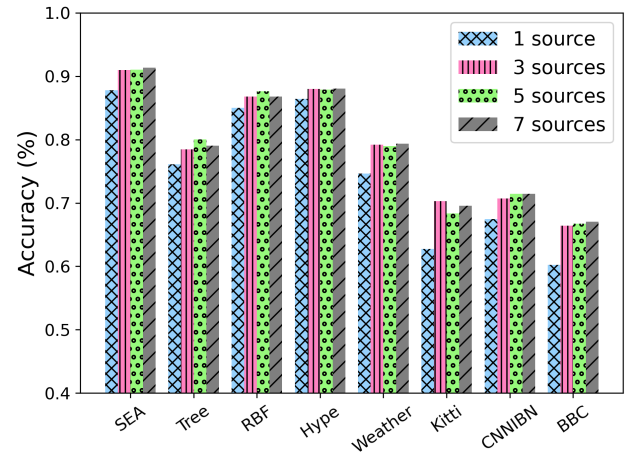


Figure 2: The influence of the different number of sources.

the covariate shift but also adaptively adjust sample weights based on the feedback from the target domain. This effectively avoids negative transfer from irrelevant data, thereby ensuring better prediction accuracy. Although AOMSDA and MCMO also consider exploiting the complementary information among multiple source data streams, they ignore the underlying correlation between various streams. In contrast, OBAL employs an adaptive re-weighting approach to iteratively decrease the weights of negative transfer samples and strengthen the weights of positive transfer samples based on the predictive feedback from the target domain. As a result, OBAL achieves the best predictive performance.

Ablation Study. To validate the rationality of each component and its impact on the overall classification results, we designed three variants of OBAL. As shown in Table 2, OBAL_{v1} as a baseline design does not consider the synchronous drift and covariate shift adaptations. In this situation, each stream is assigned a base classifier and it is updated incrementally. Thus, the performance of OBAL_{v1} is the worst, and significantly lower than that of OBAL on all datasets. This phenomenon highlights the crucial role of concept drift adaptation in dynamic environment learning. OBAL_{v2} considers the synchronous drift in each stream while ignoring the covariate shift alignment. OBAL_{v3} further employs the traditional CORAL strategy to align the covariate shift which does not explore the dynamic correlation. By comparing OBAL_{v2} and OBAL_{v3} , it can be seen aligning the covariate shift can effectively enhance the performance of the target prediction. Furthermore, the final OBAL highlights the significance of appropriate weights in mitigating the influence of irrelevant source samples and effectively addressing the problem of covariance shift.

Influence of Source Numbers. In this section, we examine the impact of the number of source streams. To ensure a fair comparison with a fixed target stream, we initially sample seven streams from all datasets and vary the number of source streams. Specifically, we evaluate the performance of OBAL using 1, 3, 5, and 7 source streams, respectively.

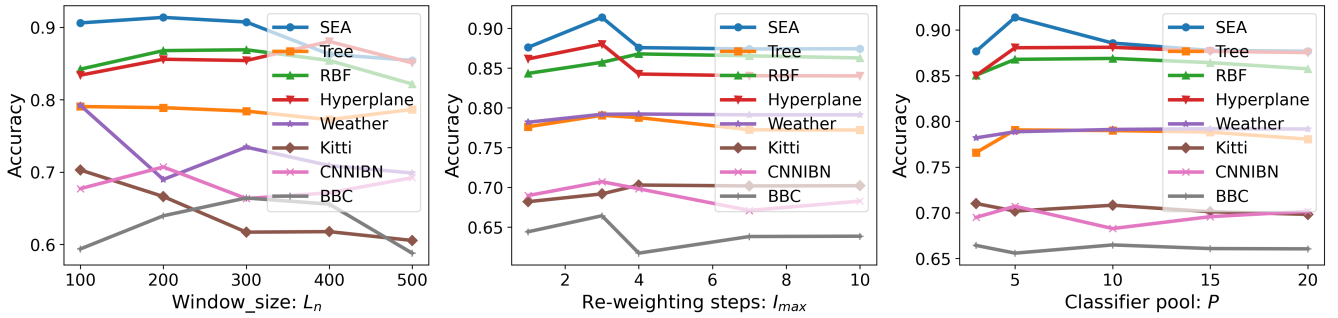


Figure 3: The effect of different parameters on classification accuracy.

This experiment first investigates whether using multi-source streams improves predictive capability compared to a single-source stream. From Figure 2, we can observe that the performance of multi-source streams outperforms single-stream performance on all datasets. This indicates that multi-source streams can provide additional complementary information to enhance predictive performance. However, as the number of source streams increases, there may be a decline in performance. For example, the performance with five source streams is better than that with seven source streams on the Tree dataset. This may be because as the number of source streams increases, the complexity of the model also increases, which affects its performance. Overall, the performance of OBAL is stable across various sources, which demonstrates that our proposed method can easily adapt to different numbers of data streams.

Parameter Sensitivity. In the proposed OBAL, there are three main parameters affecting the classification performance, including the window size of the initialization stage L_n , the re-weighting steps I_{max} , and the maximum classifier pool size $|P|$. To analyze their impact on the overall performance, we carry out experiments under various values of all parameters on all datasets. Here, we set $L_n \in \{100, 200, 300, 400, 500\}$, $I_{max} \in \{1, 3, 5, 7, 10\}$ and $|P| \in \{1, 5, 10, 15, 20\}$. During the experiment, each parameter is tuned while others are kept fixed, and the various predictive performances are shown in Figure 3.

Different datasets display varying optimal window sizes due to their unique drift frequencies and periods. For those with frequent drifts, a larger window might encompass multiple concepts, complicating accurate covariate adaptation. Hence, matching the window size to the dataset’s drift characteristics is crucial for effective prediction. In the re-weighting phase, the optimal number of iterations for most datasets is three. This is because the algorithm tends to overfit during the initialization phase with an increasing number of iterations. Additionally, as the classifier pool size grows, predictive performance generally improves across datasets, underscoring the importance of retaining historical data. However, after a certain threshold, this performance enhancement plateaus. Detailed parameter settings are shown in Table S2 in the supplementary.

	OBAL _{v1}	OBAL _{v2}	OBAL _{v3}	OBAL
SEA	79.54	82.48	88.76	90.98
Tree	72.42	74.74	77.01	78.45
RBF	79.78	81.41	84.23	86.78
Hyperplane	81.42	82.35	86.34	88.01
Weather	72.25	74.18	77.43	79.22
Kitti	62.14	64.09	68.24	70.29
CNNIBN	63.12	67.44	69.01	70.71
BBC	58.02	62.77	64.12	66.43

Table 2: Classification accuracy (%) of OBAL variants.

Time Complexity and Execution Time. As detailed in Supplementary S4, we analyze the time complexity of OBAL, where the overall complexity is given by $O(L_n \log(L_n))O(I_{max}N) + O(L_n) + O(N)O(L_n \log(L_n))$. Since N and I_{max} are both quite small, the complexity of OBAL primarily depends on the size of L_n . Therefore, we can adjust the value of L_n to execute OBAL efficiently within the available resources. Moreover, Table S3 in the Supplementary compares execution times, revealing that OBAL ranks second after Melanie, underscoring its competitive runtime.

Conclusion

In this work, we have addressed a significant gap in multistream classification, where the dynamic relationships between streams have largely been overlooked. This oversight can often result in the issue of negative transfer stemming from irrelevant data. To overcome this challenge, we introduced the Online Boosting Adaptive Learning (OBAL) method, coupled with the proposed AdaCOSA algorithm, effectively exploring the dynamic correlation among various streams. The experiments performed on several synthetic and real-world data streams have shown that our method effectively navigates the dynamic correlations between streams, mitigates covariate shifts, and adeptly handles asynchronous drift using a GMM-based weighting mechanism. The insights gained from this study not only advance the field of multistream classification but also provide a promising direction for future research in adaptive learning across various dynamic data environments.

Acknowledgments

The work presented in this paper was supported by the Australian Research Council (ARC) under Laureate project FL190100149 and discovery project DP200100700.

References

- Bifet, A.; and Gavaldà, R. 2007. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*, 443–448. SIAM.
- Chandra, S.; Haque, A.; Khan, L.; and Aggarwal, C. 2016. An adaptive framework for multistream classification. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, 1181–1190.
- Chiu, C. W.; and Minku, L. L. 2020. A diversity framework for dealing with multiple types of concept drift based on clustering in the model space. *IEEE Transactions on Neural Networks and Learning Systems*, 33(3): 1299–1309.
- Ditzler, G.; and Polikar, R. 2012. Incremental learning of concept drift from streaming imbalanced data. *IEEE transactions on knowledge and data engineering*, 25(10): 2283–2301.
- Du, H.; Minku, L. L.; and Zhou, H. 2019. Multi-source transfer learning for non-stationary environments. In *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.
- Gama, J.; Medas, P.; Castillo, G.; and Rodrigues, P. 2004. Learning with drift detection. In *Advances in Artificial Intelligence—SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29–October 1, 2004. Proceedings 17*, 286–295. Springer.
- Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.
- Gomes, H. M.; Bifet, A.; Read, J.; Barddal, J. P.; Enembreck, F.; Pfahringer, B.; Holmes, G.; and Abdesslem, T. 2017. Adaptive random forests for evolving data stream classification. *Machine Learning*, 106(9): 1469–1495.
- Haque, A.; Wang, Z.; Chandra, S.; Dong, B.; Khan, L.; and Hamlen, K. W. 2017. Fusion: An online method for multistream classification. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 919–928.
- Jiao, B.; Guo, Y.; Yang, S.; Pu, J.; and Gong, D. 2022. Reduced-space Multistream Classification based on Multi-objective Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*.
- Jothimurugesan, E.; Hsieh, K.; Wang, J.; Joshi, G.; and Gibbons, P. B. 2023. Federated learning under distributed concept drift. In *International Conference on Artificial Intelligence and Statistics*, 5834–5853. PMLR.
- Kim, Y.; and Park, C. H. 2017. An efficient concept drift detection method for streaming data under limited labeling. *IEICE Transactions on Information and systems*, 100(10): 2537–2546.
- Li, W.; Yang, X.; Liu, W.; Xia, Y.; and Bian, J. 2022. DDG-DA: Data Distribution Generation for Predictable Concept Drift Adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 4092–4100.
- Liu, A.; Lu, J.; and Zhang, G. 2020. Diverse instance-weighting ensemble based on region drift disagreement for concept drift adaptation. *IEEE transactions on neural networks and learning systems*, 32(1): 293–307.
- Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; and Zhang, G. 2018. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12): 2346–2363.
- Miyaguchi, K.; and Kajino, H. 2019. Cogra: Concept-drift-aware stochastic gradient descent for time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4594–4601.
- Oliveira, G.; Minku, L. L.; and Oliveira, A. L. 2021. Tackling virtual and real concept drifts: An adaptive gaussian mixture model approach. *IEEE Transactions on Knowledge and Data Engineering*.
- Pratama, M.; de Carvalho, M.; Xie, R.; Lughofer, E.; and Lu, J. 2019. ATL: Autonomous knowledge transfer from many streaming processes. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 269–278.
- Renchunzi, X.; and Pratama, M. 2022. Automatic online multi-source domain adaptation. *Information Sciences*, 582: 480–494.
- Song, Y.; Lu, J.; Liu, A.; Lu, H.; and Zhang, G. 2021a. A segment-based drift adaptation method for data streams. *IEEE transactions on neural networks and learning systems*, 33(9): 4876–4889.
- Song, Y.; Lu, J.; Lu, H.; and Zhang, G. 2021b. Learning data streams with changing distributions and temporal dependency. *IEEE Transactions on Neural Networks and Learning Systems*.
- Street, W. N.; and Kim, Y. 2001. A streaming ensemble algorithm (SEA) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 377–382.
- Sun, B.; Feng, J.; and Saenko, K. 2016. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.
- Vyas, A.; Kannao, R.; Bhargava, V.; and Guha, P. 2014. Commercial block detection in broadcast news videos. In *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*, 1–7.
- Wang, K.; Lu, J.; Liu, A.; Zhang, G.; and Xiong, L. 2021. Evolving gradient boost: A pruning scheme based on loss improvement ratio for learning under concept drift. *IEEE Transactions on Cybernetics*.
- Wang, Q.-F.; Geng, X.; Lin, S.-X.; Xia, S.-Y.; Qi, L.; and Xu, N. 2022a. Learngene: From open-world to your learning task. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8557–8565.

- Wang, Z.; Dai, Z.; Póczos, B.; and Carbonell, J. 2019. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11293–11302.
- Wang, Z.; Zhou, C.; Du, B.; and He, F. 2022b. Self-paced Supervision for Multi-Source Domain Adaptation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*.
- Xu, K.; Ma, Y.; Wei, B.; and Li, W. 2023. Open-Ended Diverse Solution Discovery with Regulated Behavior Patterns for Cross-Domain Adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 10585–10593.
- Yang, C.; Cheung, Y.-m.; Ding, J.; and Tan, K. C. 2021. Concept drift-tolerant transfer learning in dynamic environments. *IEEE Transactions on Neural Networks and Learning Systems*, 33(8): 3857–3871.
- Yoon, S.; Lee, Y.; Lee, J.-G.; and Lee, B. S. 2022. Adaptive Model Pooling for Online Deep Anomaly Detection from a Complex Evolving Data Stream. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2347–2357.
- Yu, E.; Song, Y.; Zhang, G.; and Lu, J. 2022a. Learn-to-adapt: Concept drift adaptation for hybrid multiple streams. *Neurocomputing*, 496: 121–130.
- Yu, H.; Zhang, Q.; Liu, T.; Lu, J.; Wen, Y.; and Zhang, G. 2022b. Meta-ADD: A meta-learning based pre-trained model for concept drift active detection. *Information Sciences*, 608: 996–1009.
- Zhou, M.; Lu, J.; Song, Y.; and Zhang, G. 2023a. Multi-Stream Concept Drift Self-Adaptation Using Graph Neural Network. *IEEE Transactions on Knowledge and Data Engineering*.
- Zhou, Z.; Guo, L.-Z.; Jia, L.-H.; Zhang, D.; and Li, Y.-F. 2023b. ODS: Test-Time Adaptation in the Presence of Open-World Data Shift.