Adaptive Hybrid Ensemble Method for Accelerate Adaptation of Concept Drift

Wenjian Wang (wjwang@sxu.edu.cn)
Shanxi University

Husheng Guo
Shanxi University

Yutong Zhang
Shanxi University

Ni Sun
Shanxi University

Research Article

Keywords: Concept drift, Adaptive learning, Local base learner, Global base learner, Hybrid ensemble

Posted Date: June 28th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3101864/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.
Adaptive hybrid ensemble method for accelerate adaptation of concept drift

Husheng Guo · Yutong Zhang · Ni Sun · Wenjian Wang

Received: date / Accepted: date

Abstract Because the insufficient new distribution training samples after concept drift occurs in streaming data, the performance of online learning model degrades and cannot quickly recover. Therefore, an adaptive hybrid ensemble method for accelerate adaptation of concept drift (AHE_A2CD) is proposed. After concept drift occurs, the proposed method extracts local information from the streaming data through the weighted base classifiers located in the classifier pool. The local information is supplemented into the current data block through expanding the data to make up for the lack of current distribution data after concept drift occurs and to build an efficient local base learner that conforms to the current data distribution. On this basis, the key data information at different stages is extracted by local base learner, and the current data is adaptively selected by the data distribution.
to construct diverse global base learner. Through the hybrid ensemble of the high-performance local base learner and the diverse global base learner, this method can adaptively learn the changing streaming data and improve the adaptability after concept drift occurs. Experimental results show that this method can accelerate the convergence of the online learning model after concept drift occurs and improve the real-time performance of streaming data classification.

**Keywords** Concept drift · Adaptive learning · Local base learner · Global base learner · Hybrid ensemble

### 1 Introduction

At present, a dynamically updated data’s explosion is generated in various practical application areas such as e-commerce, financial analysis, and cloud computing. Unlike traditional static data, these data are often generated in the form of streaming data[1-3]. Such streaming data, which is temporal, dynamic and infinite in nature, tends to change its data distribution over time, i.e. generating concept drift [4]. The concept drift existing in streaming data poses many challenges to traditional static data processing techniques and algorithms. For example, in weather forecasting, abnormal changes in climate may affect the forecast results; in the medical field, the resistance of a patient may change in response to his or her physical condition; in the fund market, the ups and downs of a fund may vary with changes in economic policies.

As a result, when concept drift occurs in streaming data, the latest data distribution is considerably different from the data distribution before concept drift. This causes the fact that the online learning model is not able to quickly adapt to the data distribution changes. Moreover, the generalization performance of the model is considerably degraded and it cannot converge to the new state quickly. Therefore, when concept drift occurs, it is important to accelerate the convergence of the online learning model so that the online learning model can quickly adapt to the new data distribution. Ensemble learning is an effective way to deal with concept drift. However, when concept drift occurs, the performance of several base classifiers in the ensemble model degrades and the overall performance of the model decreases.

To solve the problem of degradation as well as slow convergence after concept drift related to the performance of the ensemble learning model, this study proposes an adaptive hybrid ensemble method to accelerate the adaptation of concept drift and deal with the concept drift problem. By extracting the key data information from the streaming data and applying it to the ensemble model that is dynamically updated in real time, we construct a high-performance local base learner and diverse global base learners. The aim is to achieve good but different base learners in the ensemble model. Furthermore, we aim to improve the model generalization performance and enable fast convergence to
the new distribution. The main contributions of the method in this study are as follows:

(1) Hybrid ensemble model of local and global base learners is proposed to improve online learning model convergence.
(2) Base learners are built by extracting key information, which can improve diverse and effective of ensemble.
(3) This method provides a favorable approach for adaptive model updating after concept drift occurs.

The rest of this article is organized as follows. Section 2 is an overview of related work on processing streaming data with concept drift. Section 3 describes the proposed AHE_A^2CD in detail, including three parts: local base learner construction, global base learner construction and hybrid ensemble of local base learner and global base learner. Section 4 provides a theoretical proof of the effectiveness of the proposed method. Section 5 presents experiments and discussions. Conclusions are presented in Section 6.

2 Related work

Currently, there are two main strategies to improve the convergence performance of models after concept drift occurs in streaming data. One is the active acceleration strategy. It uses the detection of concept drift as a trigger by detecting changes in data distribution of streaming data or the accuracy of the classifier. Then, it accelerates the model when concept drift is detected. The other is the passive adaptive strategy, which adapts to changes in the distribution of streaming data by continuously updating the model regardless of whether concept drift occurs [5].

The active acceleration strategy has mechanisms to detect concept drift, the strategy characterises and quantifies concept drift by identifying change points or change intervals. Moreover, it will discard the current classifier and reconstruct the classifier when concept drift is detected [6]. The active acceleration strategy is divided into two steps: change detection and model construction. Change detection is performed by examining data distribution from the input, mainly through error analysis [7-8], data distribution, and multiple hypothesis testing. Typical methods in drift detection based on error analysis include the following: drift detection based on sample error rate [9-10] and sample variance [11-12]. In data distribution based drift detection, the difference between the historical data and new data distribution is generally quantified by distance function [13]. Typical methods include the following: region concept drift detection [14], least squares density difference change detection [15], and equal density estimation method [16]. Multiple hypothesis testing detects concept drift in a number of different ways [17], such as typically based on the Hoeffding inequality hypothesis [18] and maximum likelihood estimation test approach
In the model construction process, the main methods are windowing technique and weighting mechanism. They learn from new data to adapt to changes in data distribution into streaming data and discard outdated data when concept drift is detected. Windowing technique collects information through sliding windows. Furthermore, it stores the latest data to train updated models [20-21]. Typical methods include the following: dynamic incremental learning methods based on double-layer windows [22]. However, the window size is difficult to set. It is either too large or too small. The window size can affect the performance of the model. Weighting mechanism considers all available samples. Typical methods include weighting based on time [23], precision [24], and instances [25]. Weighting mechanism requires previously acquired data to be stored in memory, making it difficult to accommodate large datasets. Although active learning strategy is crucial in solving the problem of detecting concept drift, the learning process may cause problems such as slow convergence as well as degradation of real-time performance owing to missed or false detection of concept drift site.

The passive adaptive strategy focuses more on the convergence performance of the model after concept drift and updates the model directly when new data arrives. Passive adaptive strategy currently has two main approaches. One is based on single classifier updates. The other is based on ensemble classifier updates. Among single classifier-based update methods, decision tree is the most commonly used classifier in the classical very fast decision tree algorithm (VFDT) [26] and some extensions such as concept-adapting very fast decision tree (CVFDT) [27] and incrementally optimized very fast decision tree (ioVFDT) [28], which train unbalanced data through adaptive sliding windows. These models are small and have less memory occupation while having better classification accuracy. The single model is very stable and malleable. However, its structure is complex and expressive. The more current approach is the ensemble approach, characterized by the composition of multiple base classifiers as well their combination according to a certain strategy. The aim is to form a strong learner, which can reduce decision errors and obtain better prediction results than single classifier, easily coping with changes in streaming data. Typical methods include precision-based weighted ensemble methods [29] and weight-based update ensemble model methods [30]. In addition, some algorithms based on transfer learning [31], multi task learning [32-33], and deep learning [34-36] to the ensemble methods are proposed. The passive adaptive strategies enable models can achieve high classification accuracy. However, they are relying on new data to update the model during the learning process. This can afford the model that is insensitive to data with changing distributions, i.e., slower convergence of online learning models after concept drift has occurred.
The adaptive hybrid ensemble method for accelerating the adaptation of concept drift proposed in this study uses the extracted effective information to effectively compensate the lack of new distribution data after concept drift and to build base learners with good classification performance and diversity. Compared with traditional methods, the hybrid ensemble of the local base learner and the global base learner in this study, which effectively accelerates the convergence of the online learning model and improves its generalization performance.

3 Adaptive Hybrid Ensemble Method for Accelerated the Adaption of Concept Drift

In this study, we propose an adaptive hybrid ensemble method for accelerating the adaptation of concept drift, which achieves an accelerated convergence (adaptation) of online learning model after concept drift by continuously and dynamically updating the model. When concept drift occurs, the local information in the historical data is extracted by the base classifiers in the classifier pool and combined with the current data to effectively improve the performance of the local base learner constructed at the concept drift site. Based on this, various global base learners are constructed by extracting the key data information from different stages of historical data and adaptively selecting current data through local base learner to improve the overall learning performance of the ensemble model for the streaming data. The hybrid ensemble of local base learner and global base learners can consider different types of concept drift and effectively improve the generalization performance of the online learning model. The overall framework process of the adaptive hybrid ensemble method for accelerate adaptation of concept drift (AHE_A^2CD) method is shown in Fig.1.

Fig. 1: Overall framework process diagram of AHE_A^2CD method
### 3.1 Local base learner construction

Assume that there exists the streaming data $SD = \{D_1, \cdots, D_t \cdots \}$, where $D_t = \{d^1_t, \cdots, d^\omega_t\}$ denotes a subsequence of data generated in the time interval $t$ to $t+1$, with chunk size of $\omega$, $d^i_t = (x^i_t, y^i_t)_{i=1}^{\omega}$ represents one of the data in $D_t$, and $y^i_t$ is the label of the sample $x^i_t$. After the concept drift, the data distribution changes significantly and the new data distribution is small and insufficient to build a good performance classification model. Thus, the new data distribution needs to be expanded to improve the generalization performance of the local base learner.

To fully extract the newly distributed data, a pool of updatable classifiers $f_F = \{f_1, \cdots, f_k\}$ ($k$ is the upper limit of base classifiers in the pool) is maintained, which is directly trained based on different historical data blocks. Owing to the complexity and evolution of the streaming data, the classification models obtained by direct training conducted on different data blocks have different classification results i.e., the performance of base classifiers in the classifier pool is inconsistent. Therefore, weights are assigned to the base classifiers and we use them to extract the key data information from the historical samples.

When $D_t$ arrives, a real-time base classifier $f$ is learned and the testing accuracy $acc$ is obtained and the earliest base classifier in the classifier pool is removed. The base classifiers from the classifier pool are tested on $D_t$ to obtain the testing accuracy $\{acc_1, \cdots, acc_k\}$ and combined with the testing accuracy result of $f$ to normalize the weights, i.e., $\{\omega_1, \cdots, \omega_k\}$ and $\omega$.

The closer to the latest distribution in the streaming data, the simple it is to train a better base classifier. Therefore, when testing blocks of historical data separately, the more distant the blocks are from the latest distribution, the more base classifiers will be required to make decisions together. Therefore, the correct data predicted on different data blocks is extracted by real-time base classifier with an increasing number of weighted base classifiers as the key data information.

$$E(\omega_p, f_p)^k \oplus [\omega, f], D_{t-1}) \to D_{t-1}', p = k, \cdots, 2, i = 1, \cdots, k \quad (1)$$

Equation (1) shows the extraction function of the key data information, where $D_{t-1}'$ denotes the data with correct prediction, $\oplus$ denotes the ensemble of different classifiers, and $E(\cdot, \cdot)$ denotes data extraction.

The set of correct predictions obtained on different data blocks is merged with current data block $D_t$ for training to obtain local base learner $FP$ with good generalization performance, i.e.,

$$Tr(D_{t-1}' \cup D_t) \to FP, i = 1, \cdots, k \quad (2)$$
equation (2) shows the local base learner construction function. \( T_r(\cdot) \) denotes the construction of a model. Fig. 2 shows the construction process of the local base learner.

3.2 Global base learner construction

After obtaining the local base learner \( FP \), the global base learner \( FG = \{F_1, \cdots, F_k\} \) is trained on this basis. This is because \( FP \) is more reliable on \( D_t \) compared to the real-time base classifier \( f \). Using \( FP \) as a base, the first \( k \) chunks of data from \( D_{t-1} \) to \( D_{t-k-1} \) are predicted in turn. Furthermore, the data with the correct predictions is extracted using the absolute majority voting method as the overall key data, and the corresponding testing accuracy \( Acc_p \) is obtained. Also, in the prediction process, each time one of the global base learners is trained, it will be added to the subsequent decisions.

\[
E([F_p^{\text{kp}} \oplus [FP], D_{t-i} \cup \cdots \cup D_{t-k-i+1}) \rightarrow D'_p, p = k, \cdots, 1, i = 1, \cdots, k \quad (3)
\]

Equation (3) shows the overall key data information extraction process, where \( D'_p \) is the correct predicted data, i.e., the key data information.

To achieve dynamic ensemble of good global base learner in this study, a data distribution measurement parameter \( \theta \) is set up by comparing the distribution of the data with the current testing accuracy \( Acc_p \). If \( Acc_p > \theta \), it means that the data are similarly distributed and a better performing base learner can be obtained by relying on the correct data extracted alone, then the correct data as key data information is used to train \( F_p(p = k, \cdots, 1) \). If \( Acc_p < \theta \), then the data distribution is unbalanced and a base learner with good performance may not be obtained from the extracted correct data alone. Thus, the
key data information is combined with the current data block for updating $F_p$(Equation(4)). Fig.3 shows the construction process of the global base learner.

$$\begin{align*}
    &\{ Tr(D_p^r) \rightarrow F_p, Acc_p > \theta \\
    &Tr(D_p^r \cup D_t) \rightarrow F_p, Acc_p < \theta, p = k, \ldots, 1
\end{align*}$$

(4)

Fig. 3: The construction process of global base learner

3.3 Hybrid ensemble of local base learner and global base learner

Unlike traditional process for replacing the base classifier with the lowest prediction accuracy or weight, the AHE-A^3CD method enables dynamic updating of the ensemble model by updating all base learners. Accuracy tests are performed on $D_t$ to obtain the local base learner testing accuracy $Acc_P$ and the global base learner testing accuracy $Acc_G_i (i = 1, 2, \ldots, k)$, respectively, which are used as performance measures and normalized to obtain the model weights, i.e.,

$$\omega_P = \frac{Acc_P}{\sum_{i=1}^{k} Acc_G_i + Acc_P}$$

(5)

$$\omega_G_i = \frac{Acc_G_i}{\sum_{i=1}^{k} Acc_G_i + Acc_P}, i = 1, 2, \ldots, k$$

(6)

The predictive probability function at time $t$ is a weighted hybrid ensemble of the local base learner and the global base learner ($k$ is the upper limit of the base classifier in the global base learner).

$$f(x) = \omega_P \ast F_t(x) + \sum_{i=1}^{k} \omega_G_i \ast F_i(x)$$

(7)
The AHE\textsubscript{A\textsuperscript{2}CD} method continuously updates the ensemble model without the need for complex weight calculation and addition as well as deletion of base classifiers compared to traditional methods. It should yield better classification accuracy after abrupt and gradual concept drift and when the streaming data is stable. The hybrid ensemble of local base learner and global base learner can adaptively balance the sensitivity to abrupt and gradual concept drift, which is crucial in capturing the overall trend and direction of concept drift.

3.4 AHE\textsubscript{A\textsuperscript{2}CD} Algorithm

The AHE\textsubscript{A\textsuperscript{2}CD} method proposed in this study accelerates the convergence of online learning models via updating ensemble real time. By extracting valid information from streaming data to construct high-performance local base learner and diverse global base learner, the generalization performance of the model is improved and its convergence is effectively accelerated. The AHE\textsubscript{A\textsuperscript{2}CD} algorithm is shown as follows.

\textbf{Algorithm 1} AHE\textsubscript{A\textsuperscript{2}CD} algorithm

\textbf{Inputs:}
\begin{itemize}
\item $D = \{D_1, \cdots, D_t, \cdots\}$: Streaming data sequences.
\item $\omega$: Chunk size.
\item $k$: Upper limit of base classifiers in the classifier pool and global base learner.
\item $\theta$: Data distribution measurement parameter.
\end{itemize}

\textbf{Outputs:}
Hybrid ensemble of local base learner $FP$ and global base learner $FG = \{F_1, \cdots, F_k\}$.

\textbf{Process:}
\begin{enumerate}
\item \textbf{while} streaming data sequence $SD$ is not finished \textbf{do}
\item Learn the real-time base classifier $f$ on $D_t$, and obtain the testing accuracy $acc$;
\item Test on $D_t$ using the base classifiers from the classifier pool to get the testing accuracy and normalize them to get the weights;
\item Extract of local information from historical data using weighted base classifiers by equation (1);
\item Train local base learner $FP$ with the combination of local information and $D_t$ by equation (2);
\item \textbf{for} $p = 1:k$ \textbf{do}
\item $FP$ and the continuously available $F_p$ are tested against the historical data to obtain the testing accuracy $Acc_p$ and the key data information by equation (3);
\item \textbf{if} $Acc_p > \theta$ \textbf{then}
\item All key data information is combined with $D_t$ to train $F_p$ by equation (4);
\item \textbf{else}
\item All key data information is used to train $F_p$ by equation (4);
\item \textbf{end if}
\item \textbf{end for}
\item \textbf{end while}
\end{enumerate}
4 The effectiveness of adaptive hybrid ensemble analysis

This section aims to theoretically analyze the validity and rationality of the proposed AHE-A²CD method. This method accelerates the convergence of online learning models by extracting the key data information before concept drift for current data information supplementation after the concept drift. It selects the correctly classified historical data to update the local base learner and uses this updated local base learner to construct the global base learner. Because the key data information selected by the proposed method is closer to the current distribution of $D_t$ than traditional online learning. Therefore, the proposed adaptive hybrid ensemble method can effectively improve the quality of training data set after concept drift, and improve the real-time generalization performance of the model.

We denote $z = (x, y)$ and set $\varphi_1$ as the distribution before the concept drift, whose density function is $P_1(z)$, and set $\varphi_2$ is the distribution after the concept drift, whose density function is $P_2(z)$. $c(z)$ is the probability of data point $z$ that is correctly classified by the current model $f_F$. Denote $z'$ as the data selected by the current model $f_F$, whose distribution is denoted as $\varphi_3$, and the density function of $\varphi_3$ will be:

$$P_3(z') = \frac{P_1(z')c(z')}{\int P_1(z)c(z)dz} \quad (8)$$

The average accuracy of model $f_F$ over some probability density function $P(z)$ can be represented as $\int P(z)c(z)dz$. When concept drift occurs, there will be a significant difference between $P_1(z)$ and $P_2(z)$. Thus, the accuracy of $f_F$ over distribution $\varphi_2$ will be much higher than that over $\varphi_1$, i.e.,

$$\ln \int P_2(z)c(z)dz \gg \ln \int P_1(z)c(z)dz \quad (9)$$

Next we turn to prove that in the sense of KL divergence, the distribution of data selected by the current model $f_F$ is closer to $\varphi_2$ than to $\varphi_1$, i.e., $KL(P_2||P_3) < KL(P_2||P_1)$.

Proof. According to the definition of KL divergence, i.e.,

$$KL(P_2||P_1) = \int P_2(z)ln \left( \frac{P_2(z)}{P_1(z)} \right)dz$$

$$KL(P_2||P_2) = \int P_2(z)ln \left( \frac{P_2(z)}{P_1(z)/ \int P_1(z)c(z)dz} \right)dz$$

$$= \int P_2(z)ln \left( \frac{P_2(z)}{P_1(z)} \right)dz - \int P_2(z)ln \left( \frac{c(z)/ \int P_1(z)c(z)dz)}{P_1(z)} \right)dz$$

$$= KL(P_2||P_1) - \int P_2(z)ln \left( \frac{c(z)/ \int P_1(z)c(z)dz)}{P_1(z)} \right)dz$$
Note that,

\[ \int P_2(z) \ln \left( \frac{c(z)}{\int P_1(z)c(z)dz} \right) dz \]
\[ = \int P_2(z) \ln(c(z)) dz - \int P_2(z) \ln \left( \int P_1(z)c(z)dz \right) dz \]
\[ = \int P_2(z) \ln(c(z)) dz - \ln(\int P_1(z)c(z)dz) \]

Together with inequality (9) and Jensen’s inequality, we know that if the following inequality holds, then,

\[ \ln(\int P_1(z)c(z)dz) \ll \int P_2(z) \ln(c(z)) dz \leq \ln(\int P_2(z)c(z)dz) \]

Therefore,

\[ \int P_2(z) \ln \left( \frac{c(z)}{\int P_1(z)c(z)dz} \right) dz > 0 \]

then,

\[ KL(P_2||P_3) < KL(P_2||P_1) \]

End Proof.

Based on the aforementioned conclusions, using the key data information selected by the classifier pool \( f_F \), for training and updating of the ensemble learning model, is expected to result in better online learning models.

5 Experimental results and analysis

To test the effectiveness of the proposed AHE_A^2CD in accelerating the convergence of online learning model after drift, this study uses different types of concept drift datasets for experiments. The computer configuration for running the experiment is as follows: Windows 10 operating system, i5-2.70GHz core CPU, 4GB RAM, and MATLAB R2018a. The proposed method in this study is compared with DWCDS [37], HBP [38], Resnet [39-40], Highway [41], and different levels of DNNs. The deep learning comparison methods use the ReLU activation function, a fixed learning rate of 0.001, the TensorFlow framework, and the DWCDS comparison method parameters use default values.
5.1 Datasets

5.1.1 Synthetic datasets

To test the ability of the algorithm to process different types of concept drift, this study uses the streaming data generator in the Massive Online Analysis to generate five concept drift datasets with abrupt, gradual, and incremental types.

1) *Sea*: This dataset was first proposed by Street et al. in 2001, which is a classic abrupt dataset. Each sample contains three features $f_1$, $f_2$, and $f_3$, and the corresponding category is only related to the first two features. A sample is a positive class when it satisfies $f_1 + f_2 \leq \theta$. The experiment generated a dataset containing three concept drift, and the concept drift sites were 25K, 50K, and 75K.

2) *Hyperplane*: This dataset changes the orientation and position of the Hyperplane by changing the weight of the data features to simulate the incremental concept drift. A hyperplane in $d$-dimensional space is the set of points $x$ that satisfy $\sum_{i=1}^{d} \omega_i x_i = \omega_0 = \sum_{i=1}^{d} \omega_i$, here $x_i$ is the $i$th coordinate of point $x$, and $\omega_i \in [0,1]$ is the corresponding weight. When $\sum_{i=1}^{d} \omega_i x_i \geq \omega_0$, the samples are labeled positive, otherwise it is negative.

3) *RBFblips*: This dataset generates a fixed number of random centroids through random radial basis functions. Each center has a random position, a single standard deviation, class label and weight. New examples are generated by selecting a center at random. This dataset contains three abrupt concept drifts located at 25K, 50K, and 75K.

4) *LED*: This dataset contains data used to predict the numbers on the seven-segment LED display. The experiment selects data containing 24 binary attributes, and respectively generates a gradual concept drift dataset $\text{LED}_{\text{gradual}}$ and an abrupt concept drift dataset $\text{LED}_{\text{abrupt}}$. The former contains three gradual concept drift with a width of 50 instances, with concept drift sites of 25K, 50K and 75K, respectively. The latter has an abrupt concept drift at 50K.

5) *Tree*: This dataset generates data through a decision tree. The decision tree is constructed by choosing attributes at random to split, and assigning a random class label to each leaf. New samples are generated by assigning random values to attributes which then determine the class label via the tree. This dataset contains three abrupt concept drifts located at 25K, 50K and 75K.

5.1.2 Real-world datasets

In addition to synthetic datasets, this study also selects two real datasets.
1) *Electricity*: This dataset is commonly used in the field of concept drift. It contains data on electricity prices in New South Wales, Australia, affected by weather, user demand, supply conditions and seasons. The dataset contains 45,312 pieces, each of which contains 6 features and 2 categories.

2) *Covertype*: This dataset contains the forest cover of a certain area in the US Forest Service system. The data set contains 54 attributes and 7 forest cover types.

The detailed information of the datasets used in the experiments is shown in Table 1, where for the *Covertype* dataset, a random sample of 100K data is used in the experiments.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Instances</th>
<th>Features</th>
<th>Class</th>
<th>Types</th>
<th>Drift site</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sea</td>
<td>100K</td>
<td>3</td>
<td>2</td>
<td>Gradual</td>
<td>3</td>
<td>25K,50K,75K</td>
</tr>
<tr>
<td>Hyperplane</td>
<td>100K</td>
<td>10</td>
<td>2</td>
<td>Incremental</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RBFBlips</td>
<td>100K</td>
<td>20</td>
<td>4</td>
<td>Abrupt</td>
<td>3</td>
<td>25K,50K,75K</td>
</tr>
<tr>
<td>LED</td>
<td>100K</td>
<td>24</td>
<td>10</td>
<td>Abrupt</td>
<td>1</td>
<td>50K</td>
</tr>
<tr>
<td>Led_gradual</td>
<td>100K</td>
<td>20</td>
<td>10</td>
<td>Gradual</td>
<td>3</td>
<td>25K,50K,75K</td>
</tr>
<tr>
<td>Tree</td>
<td>100K</td>
<td>30</td>
<td>10</td>
<td>Abrupt</td>
<td>3</td>
<td>25K,50K,75K</td>
</tr>
<tr>
<td>Electricity</td>
<td>45.3K</td>
<td>6</td>
<td>10</td>
<td>Unknow</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Covertype</td>
<td>581K</td>
<td>54</td>
<td>7</td>
<td>Unknow</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

5.2 Parameter setting

To verify the performance of adaptive hybrid ensemble method for accelerate adaptation of concept drift, an experimental study is conducted under different parameters.

1) Basic chunk size $w$: When the chunk size is very small, it does not represent the data distribution well and cannot accurately evaluate the performance of the model. However, when chunk size is very large, the program runs very slowly and inefficiently. Therefore, different chunk size parameters with values of 200, 300, 400, 500 and 600 are used in the experiments.

2) Concept drift parameter $\eta$: To assess the effectiveness of model recovery after the abrupt type of drift, experiments are conducted through the selection of several values of $\eta$. Because $\eta$ is very small to be fully recovered to the optimal accuracy, the evaluation paradigm is affected, and $\eta$ is too large to be recovered by the model. This phenomenon is resulting in the failure of the evaluation metrics for abrupt datasets. Therefore, $\eta = 0.75$ is used in this study.
3) Threshold convergence parameter $\varepsilon$: After the gradual concept drift, to select the appropriate convergence point, this study conducts experiments by selecting multiple values of $\varepsilon$. Owing to the large value of $\varepsilon$, the convergence point is not selected appropriately. Therefore, the convergence performance of the model cannot be effectively evaluated. Thus, $\varepsilon = 0.01$ is used in the study.

4) Data distribution measurement parameter $\theta$: To compare the data distribution and achieve the purpose of dynamic ensemble of the global model, multiple $\theta$ are selected for testing, considering $\theta = 0.9$.

5) The number of base classifiers $k$: Because $k$ is very large, it is resulting in overfitting. However, very small $k$ will not reflect the advantages of the ensemble. In the experiment, we choose $k = 5$.

6) Base classifier: A library for support vector machines (Libsvm) [42] is used as the base classifier in the experiments. In addition a Gaussian kernel is used in the experiments and the default value of $g = 1/n$ is chosen for the kernel parameters, and $n$ is the dimensionality of the dataset. The penalty parameter $C = 10$. The model selection of the base classifier is not the focus of this study, and the relevant methods can be found in [43].

5.3 Evaluation indicators

To measure the convergence performance of online learning models, this study focuses on the following aspects of evaluation.

(1) $Avgracc$ (Average real accuracy): $Avgracc$ is the average real-time accuracy at each timestamp, which reflects the real-time performance of the model. It is defined as:

$$Avgracc = \frac{1}{T} \sum_{t=1}^{T} \frac{n_{t}}{n}$$

where $T$ represents the number of timestamps, $n$ is the number of samples obtained in each timestamp, and $n_{t}$ is the number of samples with the correct label prediction at the $t^{th}$ timestamp.

(2) $Cumacc$ (Cumulative Accuracy): $Cumacc$ reflects the overall performance of the model from the beginning to the current time. It is defined as:

$$Cumacc = \frac{1}{T_{t} \cdot n} \sum_{t=1}^{T_{t}} n_{t}$$

where $T_{t}$ represents the current accumulated cumulative step count.
Adaptive hybrid ensemble method for accelerate adaptation of concept drift

(3) \textit{Rob} (Robustness): For a particular algorithm A, the robustness on dataset D is defined as the ratio of the final cumulative accuracy of algorithm A on dataset D to the lowest accuracy of all methods [44], reflecting the effectiveness of the model in solving the task of learning from distributional change streaming data. It is defined as follows:

\[ \text{Rob}_A(D) = \frac{\text{Cumacc}_A(D)}{\min_{\alpha} \text{Cumacc}_\alpha(D)} \]  

(12)

where \( \text{Cumacc}_A(D) \) is the final cumulative accuracy of algorithm A on dataset D and \( \min_{\alpha} \text{Cumacc}_\alpha(D) \) is the minimum of the final cumulative accuracy of all algorithms (including algorithm A and all comparison algorithms) on dataset emphD, where the worst performing algorithm on dataset D has a robustness of 1.

(4) \textit{RSA} (Recovery speed under accuracy): An online learning model with good convergence can not only converge to the stable state of the new distribution within a short time after concept drift but also maintain the minimum of real-time error during the convergence process. For gradual and abrupt drifts, when the model is updated, the proportions of historical and new data are different. Gradual drift needs to retain more historical information, while sudden drift requires more attention to new information. Therefore, for different types of concept drift, its convergence performance is defined differently. Thus, the convergence performance of the model can be measured using the definition of \textit{RSA}:

\[ \text{RSA}_{abr} = \text{step} \times (1 - \text{Avgacc}) \]  

(13)

\[ \text{RSA}_{gra} = \frac{\text{acc}(c)}{\text{acc}(t)} \]  

(14)

Here, \( \text{RSA}_{abr} \) represents the convergence performance of the abrupt dataset, \textit{step} represents the number of steps required to restore the model to \( \eta \) times its pre-drift accuracy, and \( \eta \) is the concept drift parameter. The smaller the value of \( \text{RSA}_{abr} \), the better the model recovery performance after concept drift. \( \text{RSA}_{gra} \) represents the convergence performance of the gradual dataset, \( \text{acc}(t) \) represents the real-time accuracy of the convergence point, and \( \text{acc}(c) \) represents the real-time accuracy of the model when drift has not occurred. The smaller the value of \( \text{RSA}_{gra} \), the better the model recovery performance.

The point of convergence is defined as follows (as seen in Equation (15)). The testing results of \( n \) subsequent reference sites at the beginning site are used to determine whether the site is a convergence site or not. If the difference in accuracy between this site and the subsequent reference site is less than the given threshold and the average accuracy of the first \( n/2 \) and last \( n/2 \) reference sites is less than the threshold, then the site is a convergence site.
\[ \forall i, i \in \{1, \ldots, n\}, \]
\[ |acc_i - acc_{i+1}| < \varepsilon \land \left| \frac{2}{n} \sum_{j=1}^{n/2} acc_{t+j} - \frac{2}{n} \sum_{k=n/2+1}^{n} acc_{t+k} \right| < \varepsilon \]  \hspace{1cm} (15)

where \( \varepsilon \) is the threshold convergence parameter.

5.4 Experimental results and analysis

To effectively measure the accelerated convergence of the model after the drift, this study analyzes the performance of the online learning model in terms of \( \text{Avgracc}, \text{Cumacc}, \text{Rob}, \) and \( \text{RSA} \) by tracking the performance of the learning model after concept drift.

5.4.1 Results and analysis of \( \text{Avgracc} \)

In this study, the performance of the model at all points is reflected by the \( \text{Avgracc} \). Table 2 shows the results of \( \text{Avgracc} \) for different chunk size. Based on the experimental results, the \( \text{Avgracc} \) gradually increases and then stabilizes as \( w \) increases. The experimental results show that on all datasets, as \( w \) increases, it compensates for the lack of experimental data. Therefore, \( \text{Avgracc} \) shows an upward trend and when \( w \) increases to a certain level, the training data can represent the data distribution well; hence, \( \text{Avgracc} \) stays at a stable level. Therefore, to avoid \( w \) being too small to accurately assess performance or too oversized to reduce operating efficiency, this study uses the \( w = 500 \) vs. comparison methods for comparison.

Table 2: \( \text{Avgracc} \) results of different chunk size

<table>
<thead>
<tr>
<th>Datasets</th>
<th>200</th>
<th>300</th>
<th>400</th>
<th>500</th>
<th>600</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sea</td>
<td>0.838</td>
<td>0.841</td>
<td>0.843</td>
<td>0.844</td>
<td>0.845</td>
</tr>
<tr>
<td>Hyperplane</td>
<td>0.909</td>
<td>0.913</td>
<td>0.916</td>
<td>0.917</td>
<td>0.918</td>
</tr>
<tr>
<td>RBFBlips</td>
<td>0.782</td>
<td>0.843</td>
<td>0.852</td>
<td>0.857</td>
<td>0.858</td>
</tr>
<tr>
<td>LED</td>
<td>0.576</td>
<td>0.591</td>
<td>0.598</td>
<td>0.604</td>
<td>0.607</td>
</tr>
<tr>
<td>Led_gradual</td>
<td>0.590</td>
<td>0.605</td>
<td>0.613</td>
<td>0.617</td>
<td>0.618</td>
</tr>
<tr>
<td>Tree</td>
<td>0.522</td>
<td>0.558</td>
<td>0.560</td>
<td>0.565</td>
<td>0.575</td>
</tr>
<tr>
<td>Electricity</td>
<td>0.680</td>
<td>0.693</td>
<td>0.709</td>
<td>0.710</td>
<td>0.746</td>
</tr>
<tr>
<td>Covertype</td>
<td>0.593</td>
<td>0.607</td>
<td>0.611</td>
<td>0.617</td>
<td>0.623</td>
</tr>
</tbody>
</table>

Table 3 shows the \( \text{Avgracc} \) and the corresponding average rank of the different methods when chunk size of 500. By comparing the average rank values, AHE\_A\^2\_CD has the best performance, followed by Resnet. However, DNN16 has the worst performance. Based on the results presented in the table, AHE\_A\^2\_CD method can effectively handle both abrupt and incremental concept drift, with the best real-time accuracy on the corresponding datasets, slightly inferior to DNN4 on the abrupt dataset and to the DWCDS method on the real dataset.
The aforementioned results fully demonstrate that the AHE\_A\_CD method can adapt to the current environment more quickly and achieve good classification performance after concept drift.

Table 3: \textit{Avgracc} comparison results of different methods

<table>
<thead>
<tr>
<th></th>
<th>DNN2</th>
<th>DNN4</th>
<th>DNN8</th>
<th>DNN16</th>
<th>HBP</th>
<th>Resnet</th>
<th>Highway</th>
<th>DWCDS</th>
<th>AHE_A_CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sea</td>
<td>0.7081(9)</td>
<td>0.7155(8)</td>
<td>0.7495(4)</td>
<td>0.7441(6)</td>
<td>0.7771(2)</td>
<td>0.7684(3)</td>
<td>0.7448(5)</td>
<td>0.7253(7)</td>
<td>0.8438(1)</td>
</tr>
<tr>
<td>Hyperplane</td>
<td>0.8600(5)</td>
<td>0.8578(6)</td>
<td>0.8487(7)</td>
<td>0.7227(8)</td>
<td>0.8692(3)</td>
<td>0.8841(2)</td>
<td>0.8637(4)</td>
<td>0.6646(9)</td>
<td>0.9174(1)</td>
</tr>
<tr>
<td>RBFLips</td>
<td>0.8256(7)</td>
<td>0.9099(1)</td>
<td>0.8655(2)</td>
<td>0.4718(9)</td>
<td>0.8450(5)</td>
<td>0.8482(4)</td>
<td>0.8289(6)</td>
<td>0.5514(8)</td>
<td>0.8566(3)</td>
</tr>
<tr>
<td>LED</td>
<td>0.5868(4)</td>
<td>0.5810(4)</td>
<td>0.5311(7)</td>
<td>0.2784(9)</td>
<td>0.5692(6)</td>
<td>0.5893(2)</td>
<td>0.5796(5)</td>
<td>0.3529(8)</td>
<td>0.6048(1)</td>
</tr>
<tr>
<td>Led_gradual</td>
<td>0.5773(4)</td>
<td>0.5898(2)</td>
<td>0.5311(7)</td>
<td>0.3031(9)</td>
<td>0.5650(6)</td>
<td>0.5893(3)</td>
<td>0.5700(5)</td>
<td>0.3210(8)</td>
<td>0.6170(1)</td>
</tr>
<tr>
<td>Tree</td>
<td>0.1948(6)</td>
<td>0.2057(3)</td>
<td>0.1338(8)</td>
<td>0.1141(9)</td>
<td>0.1432(7)</td>
<td>0.2036(4)</td>
<td>0.1992(5)</td>
<td>0.3618(2)</td>
<td>0.5645(1)</td>
</tr>
<tr>
<td>Electricity</td>
<td>0.6228(6)</td>
<td>0.6231(5)</td>
<td>0.5635(8)</td>
<td>0.5154(9)</td>
<td>0.5976(7)</td>
<td>0.6317(4)</td>
<td>0.6343(3)</td>
<td>0.7069(1)</td>
<td>0.7095(2)</td>
</tr>
<tr>
<td>Covertype</td>
<td>0.4900(4)</td>
<td>0.4434(3)</td>
<td>0.3941(7)</td>
<td>0.2916(9)</td>
<td>0.4086(5)</td>
<td>0.4096(5)</td>
<td>0.4029(6)</td>
<td>0.5680(2)</td>
<td>0.6167(1)</td>
</tr>
</tbody>
</table>

Average Rank | 5.50 | 4.00 | 6.25 | 8.50 | 5.50 | 3.38 | 4.88 | 5.63 | 1.38 |

In this paper, the critical difference \((CD)\) of all methods was calculated by the Bonferroni-Dunn test to show the relative performance between the AHE\_A\_CD and the comparison methods. The performance of two classifiers is significantly different if the corresponding average rank sum differs by at least the \(CD\):

\[
CD = q_{\alpha} \sqrt{\frac{k(k+1)}{6n}}
\]

where \(q_{\alpha}\) is the critical value at significance level \(\alpha\), \(k\) is the number of the comparison methods, \(n\) is the number of datasets. Based on calculations, \(CD = 2.888\) at the 0.05 significance level. The results of the statistical analysis are shown in Fig.4. The methods in which the average rank sum is within one \(CD\) to that of AHE\_A\_CD are connected with black lines. The results show that the average real-time accuracy \textit{Avgracc} of the AHE\_A\_CD method is significantly better than that of the different levels of DNN, HBP, Highway, and DWCDS.

![Fig. 4: Comparison of \textit{Avgracc} with the Bonferroni-Dunn test](image-url)
5.4.2 Results and analysis of Cumacc

Fig. 5 shows the cumulative accuracy of the different methods at each time point for a fixed chunk size of 500. The AHE-A²CD method proposed in this study has high accuracy at the initial time point, and the accuracy of the comparison methods show a slow increasing trend. This is because AHE-A²CD does not require a high number of learning parameters to be set, allowing it to maintain a high level of accuracy in the initial stages. Moreover, the proposed method has a lower level of decline after drift than the comparison methods. This is because the AHE-A²CD method extracts the key data information from streaming data to train the model, which allows the model to converge faster after drift.

![Cumacc comparison of different methods](image)

Table 4 shows the Cumacc and the corresponding ranking of the different methods when chunk size is 500. By comparing the average rank values, AHE-A²CD
has the best performance, followed by Resnet. However, DNN16 has the worst performance. Based on the results presented in the table, AHE-A^2CD method achieves the highest accuracy on all synthetic datasets except for the RBFBlips abrupt dataset, which is inferior to DNN4 on the real dataset Covertype. This demonstrates that the method proposed in this study can efficiently handle different types of concept drift, accelerate model convergence, and reflect the advantages of hybrid ensemble of different base learners.

Table 4: Cumacc comparison results of different methods

<table>
<thead>
<tr>
<th></th>
<th>DNN2</th>
<th>DNN4</th>
<th>DNN8</th>
<th>DNN16</th>
<th>HBP</th>
<th>Resnet</th>
<th>Highway</th>
<th>DWCDS</th>
<th>AHE_A^2CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sea</td>
<td>0.7495(8)</td>
<td>0.7542(7)</td>
<td>0.7861(4)</td>
<td>0.7820(5)</td>
<td>0.8083(2)</td>
<td>0.7977(3)</td>
<td>0.7803(6)</td>
<td>0.7257(9)</td>
<td>0.8445(1)</td>
</tr>
<tr>
<td>Hyperplane</td>
<td>0.8600(5)</td>
<td>0.8580(6)</td>
<td>0.8483(7)</td>
<td>0.7230(8)</td>
<td>0.8691(3)</td>
<td>0.8840(2)</td>
<td>0.8636(4)</td>
<td>0.6683(9)</td>
<td>0.9173(1)</td>
</tr>
<tr>
<td>RBFBlips</td>
<td>0.8345(7)</td>
<td>0.9130(1)</td>
<td>0.8708(2)</td>
<td>0.5379(9)</td>
<td>0.8476(4)</td>
<td>0.8586(3)</td>
<td>0.8374(6)</td>
<td>0.5552(8)</td>
<td>0.8411(1)</td>
</tr>
<tr>
<td>LED</td>
<td>0.5869(3)</td>
<td>0.5803(4)</td>
<td>0.5305(7)</td>
<td>0.2786(8)</td>
<td>0.5693(6)</td>
<td>0.5892(2)</td>
<td>0.5797(6)</td>
<td>0.3883(9)</td>
<td>0.6043(1)</td>
</tr>
<tr>
<td>Led_gradual</td>
<td>0.5776(4)</td>
<td>0.5998(2)</td>
<td>0.5344(7)</td>
<td>0.3032(9)</td>
<td>0.5650(6)</td>
<td>0.5842(3)</td>
<td>0.5699(5)</td>
<td>0.3789(8)</td>
<td>0.6176(1)</td>
</tr>
<tr>
<td>Tree</td>
<td>0.4329(4)</td>
<td>0.4575(2)</td>
<td>0.3330(8)</td>
<td>0.3033(9)</td>
<td>0.3591(6)</td>
<td>0.4471(3)</td>
<td>0.4310(5)</td>
<td>0.3514(7)</td>
<td>0.8440(1)</td>
</tr>
<tr>
<td>Electricity</td>
<td>0.6434(6)</td>
<td>0.6450(4)</td>
<td>0.5840(8)</td>
<td>0.5735(9)</td>
<td>0.5969(7)</td>
<td>0.6447(5)</td>
<td>0.6501(3)</td>
<td>0.7105(1)</td>
<td>0.6816(2)</td>
</tr>
<tr>
<td>Covertype</td>
<td>0.6812(3)</td>
<td>0.6890(1)</td>
<td>0.6681(4)</td>
<td>0.5897(8)</td>
<td>0.6674(6)</td>
<td>0.6859(2)</td>
<td>0.6681(4)</td>
<td>0.5531(9)</td>
<td>0.6283(7)</td>
</tr>
<tr>
<td>Average Rank</td>
<td>5.00</td>
<td>3.38</td>
<td>5.88</td>
<td>8.25</td>
<td>5.00</td>
<td>2.88</td>
<td>5.00</td>
<td>7.50</td>
<td>2.50</td>
</tr>
</tbody>
</table>

Further analysis of the above results, the results of the Bonferroni-Dunn testing are shown in Fig.6, which indicates that the cumulative accuracy Cumacc of the AHE_A^2CD method is significantly better than that of DNN2, DNN8, DNN16, HBP, Highway, and DWCDS.

Fig. 6: Comparison of Cumacc with the Bonferroni-Dunn test

5.4.3 Results and analysis of Rob

To evaluate the resistance of AHE_A^2CD method to concept drift in streaming data, experiments are conducted using Rob for evaluation and analysis. Fig.7
shows $\text{Rob}$ of the AHE$_A^2$CD method compared with other methods on eight datasets, where each rectangular block indicates the robustness value of the algorithm on a particular dataset and the numbers on the columns indicate the overall $\text{Rob}$ results. The more prominent the columns are, the more resistant the model is to the concept drift, i.e., the more robust it is. The results in the figure show that the overall $\text{Rob}$ of AHE$_A^2$CD ranks first among all algorithms, with DNN4 following closely and DNN16 at the bottom in terms of $\text{Rob}$. This is because the AHE$_A^2$CD method is based on ensemble framework, which includes “superior but different” base learners in the ensemble model and extracts streaming data when training the model, ensuring that historical data is important for subsequent learning. Therefore, it is very resistant to changes in the data distribution when concept drift occurs in the streaming data.

![Fig. 7: $\text{Rob}$ comparison results under different methods](image)

5.4.4 Results and analysis of RSA

To evaluate the convergence speed of the proposed method when concept drift occurs, the experiment compares and analyzes the $\text{RSA}$ of the comparison methods. The training samples obtained at current timestamp are used to test the model and are used for the training model. In this study, the newly distributed data is used to test the model when concept drift occurs, which can effectively reflect the adaptability of the model to the new data distribution. Different models have different abilities to represent the data distribution. Thus, the model may not recover the accuracy before concept drift. Therefore, this study uses recovery speed under the accuracy $\text{RSA}$ to evaluate the model.

Table 5 shows the recovery performance of the different methods on the corresponding concept drift sites on five datasets. For two gradual datasets Sea
and LED, AHE₂CD method achieves better results. This is because the AHE₂CD method enables model update by extracting the correct data across space to train the model, allowing the ensemble model to converge quickly. For the abrupt datasets, AHE₂CD method performs slightly worse than DWCDS because DWCDS will recover to higher accuracy directly after drift and then remain stable, whereas AHE₂CD gradually recovers to higher accuracy (the highest real-time accuracy is higher than that of DWCDS) and then remains stable. However, by comparison of the average rank, AHE₂CD converges better at the drift sites in most of the datasets. The aforementioned experimental results show that AHE₂CD is less affected by concept drift after it occurs and the model can be better recovered, which indicates that the AHE₂CD method proposed in this study has obvious advantages.

Table 5: RSA of different methods

<table>
<thead>
<tr>
<th></th>
<th>DNN2</th>
<th>DNN4</th>
<th>DNN8</th>
<th>DNN16</th>
<th>HBP</th>
<th>Resnet</th>
<th>Highway</th>
<th>DWCDS</th>
<th>AHE₂CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sea</td>
<td>1.20</td>
<td>1.08</td>
<td>1.00</td>
<td>1.27</td>
<td>1.23</td>
<td>1.27</td>
<td>1.16</td>
<td>1.08</td>
<td>1.05</td>
</tr>
<tr>
<td>LED</td>
<td>0.96</td>
<td>0.96</td>
<td>0.94</td>
<td>1.00</td>
<td>1.02</td>
<td>0.97</td>
<td>0.97</td>
<td>0.82</td>
<td>0.93</td>
</tr>
<tr>
<td>RGBBlips</td>
<td>1.11</td>
<td>1.11</td>
<td>1.00</td>
<td>1.13</td>
<td>1.08</td>
<td>1.02</td>
<td>0.96</td>
<td>1.23</td>
<td>1.00</td>
</tr>
<tr>
<td>Ledin</td>
<td>2.91</td>
<td>0.33</td>
<td>2.74</td>
<td>11.51</td>
<td>4.73</td>
<td>2.51</td>
<td>2.58</td>
<td>0.17</td>
<td>0.27</td>
</tr>
<tr>
<td>R/FBlips</td>
<td>4.03</td>
<td>1.84</td>
<td>4.24</td>
<td>2.41</td>
<td>3.92</td>
<td>3.60</td>
<td>3.85</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>Tree</td>
<td>26.2</td>
<td>4.60</td>
<td>5.12</td>
<td>9.42</td>
<td>1.35</td>
<td>2.95</td>
<td>2.92</td>
<td>0.35</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Table 5: RSA of different methods

- ': Not restored to pre-drift \( \eta \) times level
<table>
<thead>
<tr>
<th></th>
<th>DNN2</th>
<th>DNN4</th>
<th>DNN8</th>
<th>DNN16</th>
<th>HBP</th>
<th>Resnet</th>
<th>Highway</th>
<th>DWCDS</th>
<th>AHE₂CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sea</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.27</td>
<td>1.27</td>
<td>1.27</td>
<td>1.16</td>
<td>1.08</td>
<td>1.05</td>
</tr>
<tr>
<td>LED</td>
<td>0.96</td>
<td>0.96</td>
<td>0.94</td>
<td>1.00</td>
<td>1.02</td>
<td>0.97</td>
<td>0.97</td>
<td>0.82</td>
<td>0.93</td>
</tr>
<tr>
<td>RGBBlips</td>
<td>1.11</td>
<td>1.11</td>
<td>1.00</td>
<td>1.13</td>
<td>1.08</td>
<td>1.02</td>
<td>0.96</td>
<td>1.23</td>
<td>1.00</td>
</tr>
<tr>
<td>Ledin</td>
<td>2.91</td>
<td>0.33</td>
<td>2.74</td>
<td>11.51</td>
<td>4.73</td>
<td>2.51</td>
<td>2.58</td>
<td>0.17</td>
<td>0.27</td>
</tr>
<tr>
<td>R/FBlips</td>
<td>4.03</td>
<td>1.84</td>
<td>4.24</td>
<td>2.41</td>
<td>3.92</td>
<td>3.60</td>
<td>3.85</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>Tree</td>
<td>26.2</td>
<td>4.60</td>
<td>5.12</td>
<td>9.42</td>
<td>1.35</td>
<td>2.95</td>
<td>2.92</td>
<td>0.35</td>
<td>0.20</td>
</tr>
</tbody>
</table>

6 Conclusion and future work

To solve the problem of reduced model convergence after concept drift in streaming data, this study proposes an adaptive hybrid ensemble method for accelerate adaptation of concept drift. The proposed method compensates for the lack of new distribution samples after concept drift by extracting valid data information from streaming data and constructs high performance local base learner and diverse global base learner. The hybrid ensemble of different types of base learners improves the adaptability of the model to changing streaming data after concept drift and accelerates the convergence of the online learning model. In future work, the number of base learners in the model will
be automatically adjusted in accordance with the changing data distribution of the streaming data.

Acknowledgements This research was partially supported by the National Natural Science Foundation of China (Nos. 62276157, U21A20513, 62076154, 61503229), Special Foundation from the Central Finance to Support the Development of Local University (YDZX2020140001224), Natural Science Foundation of Shanxi Province (No. 201901D111033).

References

Husheng Guo received the Ph.D. degree from Department of Computer and Information Technology, Shanxi University in 2014. Now he is associate professor at School of Computer and Information Technology, Shanxi University, and serves as a Ph.D. supervisor in Computer Science and Technology. He has published more than 30 academic papers on machine learning and data mining. His current research interests include data mining, machine learning, kernel methods, etc.

Yutong Zhang received the Bachelor degree from College of Information, Business College of Shanxi University, China, in 2021. Now he is Master at School of Computer Science and Technology, Shanxi University. His current research interests include data mining and online learning.

Ni Sun received the Bachelor degree from College of Information, Shanxi Agricultural University, China, in 2020. Now she is Master at School of Computer Science and Technology, Shanxi University. Her current research interests include data mining and online learning.

Wenjian Wang received the B.S. degree in computer science from Shanxi University, China, in 1990, the M.S. degree in computer science from Hebei Polytechnic University, China, in 1993, and Ph.D. degree in applied mathematics from Xian Jiao Tong University, China, in 2004. She worked as a research assistant at the Department of Building and Construction, The City University of Hong Kong from May 2001 to May 2002. She has been with the Department of Computer Science at Shanxi University since 1993, where she was promoted as Associate Professor in 2000 and as Full Professor in 2004, and now serves as a Ph.D. supervisor in Computer Application Technology and System Engineering. She has published more than 80 academic papers on machine learning, computational intelligence, and data mining. Her current research interests include data mining, machine learning, deep learning and kernel methods, etc.
Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Highlights.pdf