计 算 机 研 究 与 发 展
Journal of Computer Research and Development

DOI: 10.7544/issn1000-1239.202330452
61(7): 1799−1811, 2024

面向不同类型概念漂移的两阶段自适应集成学习方法

郭虎升[1,2]　张　洋[1]　王文剑[1,2]

[1]（山西大学计算机与信息技术学院　太原　030006）

[2]（计算智能与中文信息处理教育部重点实验室（山西大学）　太原　030006）

（guohusheng@sxu.edu.cn）

Two-Stage Adaptive Ensemble Learning Method for Different Types of Concept Drift

Guo Husheng[1,2], Zhang Yang[1], and Wang Wenjian[1,2]

[1]（*School of Computer and Information Technology*, *Shanxi University*, *Taiyuan* 030006）

[2]（*Key Laboratory of Computational Intelligence and Chinese Information Processing* (*Shanxi University*), *Ministry of Education*, *Taiyuan* 030006）

Abstract　In the era of big data, there is a large amount of streaming data emerging. Concept drift, as the most typical and difficult problem in streaming data mining, has received increasing attention. Ensemble learning is a common method for handling concept drift in streaming data. However, after drift occurs, learning models often cannot timely respond to the distribution changes of streaming data and cannot effectively handle different types of concept drift, leading to the decrease in model generalization performance. Aiming at this problem, we propose a two-stage adaptive ensemble learning method for different types of concept drift (TAEL). Firstly, the concept drift type is determined by detecting the drift span. Then, based on different drift types, a "filtering-expansion" two-stage sample processing mechanism is proposed to dynamically select appropriate sample processing strategy. Specifically, during the filtering stage, different non-critical sample filters are created for different drift types to extract key samples from historical sample blocks, making the historical data distribution closer to the latest data distribution and improving the effectiveness of the base learners. During the expansion stage, a block-priority sampling method is proposed, which sets an appropriate sampling scale for the drift type and sets the sampling priority according to the size proportion of the class in the current sample block to which the historical key sample belongs. Then, the sampling probability is determined based on the sampling priority, and a subset of key samples is extracted from the historical key sample blocks according to the sampling probability to expand the current sample block. This alleviates the class imbalance phenomenon after sample expansion, solves the underfitting problem of the current base learner and enhances its stability. Experimental results show that the proposed method can timely respond to different concept drift types, accelerate the convergence speed of online ensemble models after drift occurs, and improve the overall generalization performance of the model.

Key words　streaming data; concept drift; ensemble learning; drift type; filtering stage; expansion stage

摘　要　大数据时代，流数据大量涌现. 概念漂移作为流数据挖掘中最典型且困难的问题，受到了越来越广泛的关注. 集成学习是处理流数据中概念漂移的常用方法，然而在漂移发生后，学习模型往往无法对流

收稿日期：2023-06-05；**修回日期**：2023-10-09

基金项目：国家自然科学基金项目（62276157, U21A20513, 62076154, 61503229）；山西省重点研发计划项目（202202020101003）

This work was supported by the National Natural Science Foundation of China（62276157, U21A20513, 62076154, 61503229）and the Key Research and Development Program of Shanxi Province (202202020101003).

通信作者：王文剑（wjwang@sxu.edu.cn）

数据的分布变化做出及时响应，且不能有效处理不同类型概念漂移，导致模型泛化性能下降.针对这个问题，提出一种面向不同类型概念漂移的两阶段自适应集成学习方法（two-stage adaptive ensemble learning method for different types of concept drift, TAEL）.该方法首先通过检测漂移跨度来判断概念漂移类型，然后根据不同漂移类型，提出“过滤-扩充”两阶段样本处理机制动态选择合适的样本处理策略.具体地，在过滤阶段，针对不同漂移类型，创建不同的非关键样本过滤器，提取历史样本块中的关键样本，使历史数据分布更接近最新数据分布，提高基学习器有效性；在扩充阶段，提出一种分块优先抽样方法，针对不同漂移类型设置合适的抽取规模，并根据历史关键样本所属类别在当前样本块上的规模占比设置抽样优先级，再由抽样优先级确定抽样概率，依据抽样概率从历史关键样本块中抽取关键样本子集扩充当前样本块，缓解样本扩充后的类别不平衡现象，解决当前基学习器欠拟合问题的同时增强其稳定性.实验结果表明，所提方法能够对不同类型的概念漂移做出及时响应，加快漂移发生后在线集成模型的收敛速度，提高模型的整体泛化性能.

关键词 流数据；概念漂移；集成学习；漂移类型；过滤阶段；扩充阶段

中图法分类号 TP18

近些年来，流数据在网络安全、智慧城市、气象预测等多个领域大量涌现.流数据作为一种重要的数据类型，具有持续产生、实时性强、规模巨大且数据分布动态变化等复杂特性，这给流数据挖掘任务带来了极大挑战[1-5].概念漂移是指随着时间推移或数据分布发生变化，样本的输入特征和输出标签之间的关系也发生改变的现象[6-9].此时集成模型由于没有及时学习到新的数据分布特征从而导致性能会下降.

基于集成学习的方法[10-12]利用历史数据构建基学习器，并借助特定的投票机制（如加权平均、组合投票等）进行集成决策，以此得到比单一基学习器更好的效果，解决了单一基学习器在流数据挖掘中不能把握全局信息的问题，因此利用集成学习处理概念漂移是一种有效可行的手段.然而，传统集成学习方法在漂移发生后不能对新数据分布及时做出响应，且通常认为历史数据不再适用，如果这些数据中含有对当前模型学习有帮助的样本知识，直接丢弃则会造成已有资源的浪费.此外，流数据分布变化方式的多样性易产生不同类型的概念漂移（如突变型和渐变型），不同类型漂移的数据分布变化跨度、变化快慢、变化方式等都不相同[13]，然而多数在线集成模型只关注单一类型，不能针对漂移类型进行自适应建模.

为解决上述问题，本文提出一种面向不同类型概念漂移的两阶段自适应集成学习方法（two-stage adaptive ensemble learning method for different types of concept drift, TAEL）.该方法从解决不同类型的概念漂移问题入手，检测漂移跨度以确定漂移类型，并构建了针对类型的“过滤-扩充”两阶段样本处理机制.一方面在样本过滤过程中，根据漂移类型创建非关键样本过滤器，过滤掉历史样本中的非关键因素，保证剩余的历史关键样本块的数据分布更加接近当前数据分布；另一方面在样本扩充过程中，根据漂移类型确定合适的抽样规模，由当前数据块中各个类别的规模占比设置历史关键样本的抽样优先级，并确定抽样概率，按照抽样概率进行分块优先抽样，以扩充当前样本块，为当前样本块补充样本特征的同时缓解了块内类分布不平衡.本文工作的主要贡献有3方面：

1）通过检测漂移跨度确定概念漂移类型，为不同类型漂移的自适应集成建模提供了一种可行方案；

2）通过对历史数据中非关键样本的过滤，使更新后的历史数据分布更接近最新数据分布，提高了历史基学习器的有效性；

3）通过对当前数据的扩充，缓解了当前基学习器的欠拟合问题，提高了基学习器的稳定性.

1 相关工作

目前，对含概念漂移的流数据挖掘的处理策略主要包括基于实例选择的方法和基于集成学习的方法.基于实例选择的方法通常使用滑动窗口技术来实现，其基本思想是将数据流分成固定大小的窗口，通过窗口的向前滑动来实现对概念漂移的检测和处理.ADWIN[14]通过计算子窗口之间的均值差异来判断是否发生了概念漂移.DDM[15]通过持续监视窗口内的数据样本分类错误率来检测概念漂移.STEPD[16]

通过比较最近窗口和整个窗口来检测错误率变化. DWCDS[17]提出一种双窗口机制来周期性地检测概念漂移，并对模型进行动态更新以适应概念漂移. CDTW[18]首先创建2个分别加载历史数据和当前数据的基础节点时序窗口，通过比较二者包含数据的分布变化情况来检测概念漂移. CDT_MSW[19]由单个基本滑动窗口和单个基本静态窗口来检测概念漂移.

使用集成学习处理含概念漂移流数据的研究已经取得了很多成果和进展，基于集成学习的方法大体可分为2类：在线集成和基于数据块的集成.

在线集成是一种对样本进行逐一处理的增量学习方法. 基于单样本的增量模型方法[20]首先初始化一组基分类器，使用每个时间戳下到达的单个样本更新集成模型，然后对基分类器进行加权组合. DOED[21]通过维护低多样性和高多样性的在线加权集成，从而准确地处理各种类型的漂移. 基于混合标记策略的在线主动学习集成框架[22]由一个长期固定分类器和多个动态分类器组成来适应概念漂移. CBCE[23]为每个类维护一个基学习器，并在有新样本时更新基学习器. 在线集成学习方法能够有效提高模型的实时泛化性能，但由于需要逐一处理样本，增加了计算资源，易导致学习效率较低.

基于数据块的集成是一种对固定数量的输入实例进行处理的方法. SEA[24]在连续的数据块上构建基分类器，并且使用启发式替换策略组合成固定大小的集成模型. DWMIL[25], ACDWM[26]为每个数据块创建一个基学习器，通过根据基学习器在当前数据块上的分类性能进行动态加权集成. SRE[27]在基于块的框架中保留一部分先前少数样本以平衡当前块的类分布. DUE[28]为每个数据块创建若干候选分类器，对其进行分段加权，并通过动态调整分类器权重来解决概念漂移问题. SEOA[29]将神经网络的不同层次作为基分类器进行集成，根据各基分类器在当前数据块上的决策损失进行动态加权，以实现稳定性与适应性的平衡. 然而，划分的数据块的大小通常会影响模型的性能和训练速度，因此，选择合适的数据块大小很重要.

与传统方法相比，本文提出的TAEL方法能够充分利用新旧样本信息，根据漂移类型针对性地采用两阶段样本处理机制更新历史样本块和当前样本块，实现了集成模型在概念漂移发生后对新数据分布的快速响应.

2 面向不同类型概念漂移的两阶段自适应集成

本文提出的TAEL方法的模型总体结构如图1所示. 在漂移类型检测阶段，通过检测漂移跨度 *span* 确定漂移类型. 在两阶段自适应集成阶段，首先根据漂移类型创建非关键样本过滤器 F，过滤掉历史样本集 D 上的非关键样本，然后对剩余的历史关键样本 $\hat{D}$ 进行分块优先抽样 *Sampling*，根据漂移类型确定合适的抽样规模 M，并根据样本所属类在当前样本块的规模占比设置抽样优先级 α，由 α 获得抽样概率 P，按照 P 抽取一定规模的关键样本子集 $\widetilde{D}$ 来扩充当前数据集 D_t. 在更新后的历史样本集和当前样本集中训练得到具有更高有效性的基学习器，提升了集成模型的实时泛化性能.

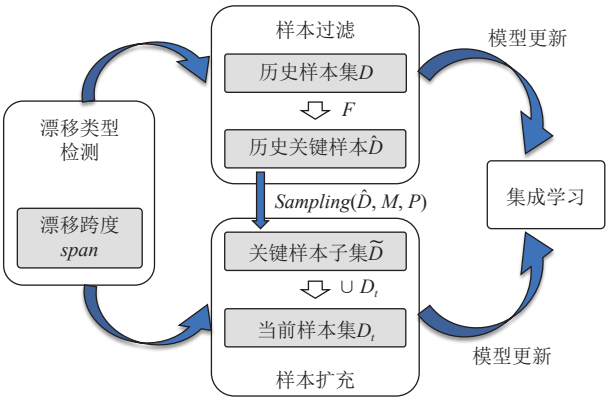


Fig. 1 The overall structure of the TAEL model

图1 TAEL模型总体结构图

2.1 漂移类型检测

流数据是指实时、连续、无限、随时间不断变化的数据序列，时刻 t 到达的样本由具有联合概率分布 $P_t(\boldsymbol{x},y)$ 的数据源产生. 在流数据挖掘任务中，样本分布的不稳定和动态变化等因素导致流数据中隐含的目标概念发生改变，即概念漂移，其本质可看作流数据的联合概率分布发生变化：

$$P_{t-1}(\boldsymbol{x},y) \neq P_t(\boldsymbol{x},y). \tag{1}$$

为了根据不同类型漂移有针对性地更新集成模型，首先在概念漂移位点处进行漂移类型检测. 本文通过计算 *span* 来检测漂移类型. *span* 由漂移开始位点和漂移结束位点间相距的时间跨度确定. 本文判断漂移是否结束的依据是后序数据分布是否已经稳定. 已知漂移开始位点 a，选取该位点后序的 L 个连续数据块 $D_{a+1}, D_{a+2}, \cdots, D_{a+L}$，在这些数据块上训练得到基学习器 $f_{a+1}, f_{a+2}, \cdots, f_{a+L}$，并得到在当前数据块上的实时预测精度 $acc_{a+1}, acc_{a+2}, \cdots, acc_{a+L}$. 计算实时预

测精度的方差:

$$s^2=\frac{\sum_{l=1}^{L}(acc_{a+l}-\overline{acc})^2}{L},\tag{2}$$

其中$\overline{acc}$为实时预测精度的平均值. s^2反映了L个基学习器的预测差异, 同时反映出位点a的后序数据分布的稳定程度. 若$s^2<\delta$(δ为漂移稳定性参数), 则认为位点$a+1$为漂移结束位点, $span=1$; 若$s^2\geqslant\delta$, 则认为漂移仍未结束, 接着从位点$a+1$开始继续上述操作, 直到得到漂移结束位点b, $span=b-a$; 若$span>\theta$(θ为漂移类型参数), 则判定此次漂移为渐变型, 否则判定此次漂移为突变型. 漂移类型检测过程如图 2 所示.

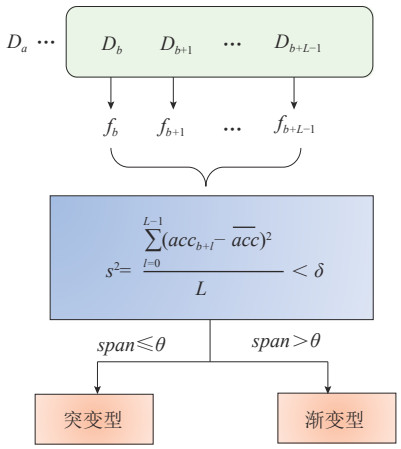


Fig. 2 Drift type detection process

图 2 漂移类型检测过程

2.2 “过滤-扩充”两阶段自适应集成学习

为充分利用当前漂移场景的样本信息和有选择地利用历史样本信息以提高集成模型在概念漂移发生后对新数据分布的适应性, 本文提出“过滤-扩充”两阶段自适应集成学习方法. 过滤阶段通过过滤非关键样本以帮助历史样本块筛选接近当前数据分布的关键样本, 扩充阶段通过向当前数据集补充过滤后保留的历史关键样本以弥补其缺少的样本特征.

2.2.1 样本过滤策略

由于历史样本中含有大量样本信息, 而非关键样本会导致该数据集上模型的有效性降低. 为“筛掉”这些无用信息, 提高样本质量, 使历史数据分布更接近当前数据分布, 本文提出一种样本过滤策略, 通过创建非关键样本过滤器F过滤掉历史非关键样本. 考虑到不同类型的概念漂移场景下数据分布的变化方式和特点不同, 因此需创建不同的F.

假设有历史数据块$D=\{D_1,D_2,\cdots,D_n\}$, 第i个数据块为$D_i=\{(\boldsymbol{x}_{ij},y_{ij})|j=1,2,\cdots,k\}$($k$为数据块大小), 由$D_i$训练得到历史基学习器$f_i$. 当前数据块$D_t=\{(\boldsymbol{x}_{tj},y_{tj})|j=1,2,\cdots,k\}$, 在$D_t$上训练得到当前基学习器$f_t$. 候选基学习器池$Q$用来存储参与集成的候选基学习器, 最大容量$s$=15.

当发生突变型概念漂移时, 数据分布急速变化, 历史数据分布和当前数据分布差异较大, 大量历史样本成为阻碍模型学习的负面因素, 导致历史基学习器的性能快速下降. 由于当前基学习器f_t在最新数据块D_t上训练得到, 反映了流数据的最新分布, 因此, 为了快速过滤掉历史非关键样本, 本文针对这种类型的概念漂移采用一种直接式过滤器, 将f_t作为每个历史数据块的非关键样本过滤器F, 即$F=f_t$. 以f_t对D_i的预测观察结果作为样本过滤条件C_i, 表达式为:

$$C_i:y_{ij}\neq F(\boldsymbol{x}_{ij}),\tag{3}$$

真实标签与f_t预测结果不同的样本将被直接过滤掉.

当发生渐变型概念漂移时, 数据分布变化较缓慢, 历史数据分布与当前数据分布虽有差异但仍相似, 历史数据块中可能只有少量样本变得非关键, 因此与突变型概念漂移的直接过滤方式不同, 渐变型概念漂移采用一种叠加式过滤器, 即通过历史数据块的后序基学习器和f_t的加权组合来叠加过滤效果, 确保充分利用历史样本知识和当前样本知识帮助进行更加准确的过滤操作. 为了实现对样本知识的有效利用, 首先需要区分每个基学习器的重要程度, 本文将基学习器在D_t上的实时预测精度作为其权重. 在此基础上, D_i的叠加过滤器F_i为:

$$F_i=\sum_{p=i+1}^{n}\frac{w_p}{\sum_{q=i+1}^{n}w_q+w_t}f_p+\frac{w_t}{\sum_{q=i+1}^{n}w_q+w_t}f_t,\tag{4}$$

$$w_g=\frac{1}{k}\sum_{j=1}^{k}[\![f_g(\boldsymbol{x}_{tj})=y_{tj}]\!],\quad g=1,2,\cdots,n,\tag{5}$$

$$w_t=\frac{1}{k}\sum_{j=1}^{k}[\![f_t(\boldsymbol{x}_{tj})=y_{tj}]\!],\tag{6}$$

其中当$[\![\cdot]\!]$中的条件成立时值为 1, 否则为 0. 以F_i对历史样本的预测观察结果作为样本过滤条件, 表达式为:

$$C_i:y_{ij}\neq F_{\mathrm{i}}(\boldsymbol{x}_{ij}),\tag{7}$$

真实标签与F_i预测结果不同的样本将被过滤掉.

经过上述操作, 符合过滤条件的样本被丢弃, 剩下更符合当前数据分布的历史关键样本块$\hat{D}_1,\hat{D}_2,\cdots,\hat{D}_n$. 由于在突变型概念漂移发生后, 过滤的样本通常较多, 训练样本不足易导致模型训练不充分, 因此本文向过滤后的每个历史样本块中补充D_t. 最后, 在更新后的历史关键样本块上训练得到$\hat{f}_1,\hat{f}_2,\cdots,\hat{f}_n$, 提高了

基学习器的有效性.

2.2.2 样本扩充策略

概念漂移发生后，当前基学习器往往欠拟合，而历史样本恰恰可以帮助当前样本集弥补其缺少的样本知识. 因此，本文提出一种样本扩充策略，将过滤后保留的历史关键样本块$\hat{D}_1,\hat{D}_2,\cdots,\hat{D}_n$用来扩充$D_t$. 然而，即使历史样本集已过滤掉部分样本，全部扩充到D_t所花费的时间代价仍较大，为解决这个问题，本文从各个历史数据块中抽取子集$\widetilde{D}_1,\widetilde{D}_2,\cdots,\widetilde{D}_n$用来扩充$D_t$. 由于扩充后的$D_t$可能存在类别不平衡，造成这种情况的原因有2种：一种原因是D_t本身就存在类别不平衡的问题，而抽取的样本子集没有改善甚至加重了这种不平衡；另一种原因是D_t本身类分布平衡，但扩充导致了类别不平衡. 因此本文从抽取方式入手，为了降低扩充后的D_t的类不平衡率，提出一种分块优先抽样方法，该方法根据样本所属类在D_t中总类别的规模占比确定抽样优先级α，由此计算得到抽样概率P，按照抽样概率P依次从各个历史关键样本块中不放回地抽取一定数量的关键样本子集用于扩充.

抽样规模的设置直接关系实验结果的好坏. 如果抽样规模太小，将会导致抽样样本不足以提供足够的关键信息；如果抽样规模太大，将会浪费时间和资源，从而降低效率. 由于突变型漂移前后数据分布的差异较大，历史关键样本往往较少，设置总抽样规模M为较小值；渐变型漂移前后数据分布间虽有差异但仍相似，历史关键样本往往较多，设置M为较大值. 因此，可将总抽样规模M和漂移跨度$span$联系起来，表达式为：

$$M=\lambda\times\frac{span}{span+1}\times\sum_{i=1}^{n}z_i, \tag{8}$$

其中λ为样本规模因子，z_i为历史数据块$\hat{D}_i$的大小. 在确定M后，$\hat{D}_i$的抽样规模M_i由其大小确定，同时为了保证有相对足够的采样样本，限制最小的块抽样规模，表达式为：

$$M_i=\max\left\{\lambda\times\frac{span}{span+1}\times z_i,\ \frac{1}{10n}\sum_{j=1}^{n}z_j\right\}. \tag{9}$$

为了缓解D_t在扩充后的类别不平衡现象，每个样本被抽中的概率与其所属类在D_t中的规模占比密切相关，即越少的类被选中的概率越大，越多的类被选中的概率越小. 因此，为历史样本中类别规模占比较小的样本设置较高的优先级，为类别规模占比较大的样本设置较低的优先级. 如果判断$\boldsymbol{x}_{ij}$所属类别为c'，设置其抽样优先级为

$$\alpha_{ij}=\begin{cases}\ln\left(\dfrac{\sum\limits_{c\in C}\sum\limits_{x=1}^{k}[\![y_{tx}=c]\!]}{\sum\limits_{x=1}^{k}[\![y_{tx}=c']\!]}\right), & c'\in C\text{且}|C|>1,\\ \ln\left(\dfrac{\sum\limits_{c\in C}\sum\limits_{x=1}^{k}[\![y_{tx}=c]\!]}{2}\right), & c'\in C\text{且}|C|=1,\\ \ln\left(\sum\limits_{c\in C}\sum\limits_{x=1}^{k}[\![y_{tx}=c]\!]\right), & \text{其他},\end{cases} \tag{10}$$

其中C为当前样本块中出现的样本类别. 抽样优先级和抽样概率成正比，$\boldsymbol{x}_{ij}$的抽样概率可表示为

$$P_{ij}=Pr((\boldsymbol{x}_{ij},y_{ij})\in\widetilde{D}_i|(\boldsymbol{x}_{ij},y_{ij})\in\hat{D}_i)=\frac{\alpha_{ij}}{\sum\limits_{p=1}^{z_i}\alpha_{ip}}. \tag{11}$$

显然，当$\hat{D}_i$中每个样本的抽样优先级相等时，有

$$P_{ij}=Pr((\boldsymbol{x}_{ij},y_{ij})\in\widetilde{D}_i|(\boldsymbol{x}_{ij},y_{ij})\in\hat{D}_i)=\frac{1}{z_i}, \tag{12}$$

分块优先抽样过程变为简单随机抽样. 将历史数据块$\hat{D}_i$的优先抽样函数表示为

$$\widetilde{D}_i=Sampling(\hat{D}_i,M_i,P_i). \tag{13}$$

依次从$\hat{D}_1,\hat{D}_2,\cdots,\hat{D}_n$中抽取数量为$M_1,M_2,\cdots,M_n$的关键样本子集$\widetilde{D}_1,\widetilde{D}_2,\cdots,\widetilde{D}_n$，将关键样本子集扩充到$D_t$中，得到扩充后的$\hat{D}_t=\widetilde{D}_1\cup\widetilde{D}_2\cup\cdots\cup\widetilde{D}_n\cup D_t$. 经过上述操作，向$\hat{D}_t$中补充了历史有用信息并且使类分布更加均衡，在扩充后的$\hat{D}_t$上训练得到的$\hat{f}_t$具有更丰富的样本特征，解决了当前基学习器的欠拟合问题，同时提高了基学习器的稳定性. 突变型和渐变型场景下的两阶段自适应集成过程如图3所示.

在将$\hat{f}_t$存储到Q前，需要判断Q是否达到最大容量s. 如果$n\geqslant s$，那么用$\hat{f}_t$替换掉在D_t上实时预测精度最小的历史基学习器：

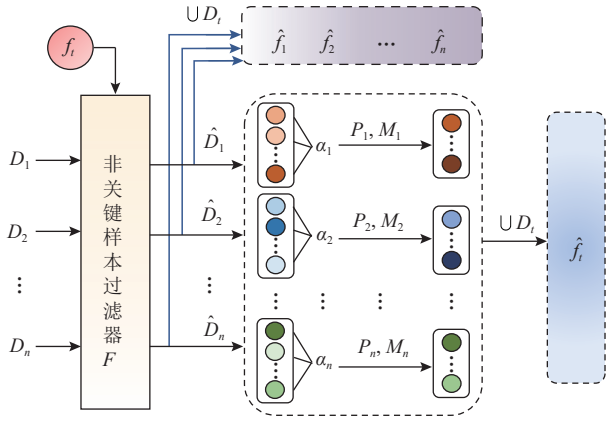
$$\hat{f}_t\rightarrow\arg\max_{\hat{f}_i\in Q}\sum_{j=1}^{k}[\![\hat{f}_i(\boldsymbol{x}_{tj})\neq y_{tj}]\!]. \tag{14}$$

最终的强分类器H对于$\boldsymbol{x}$的预测结果为多数更新后的基学习器预测的结果，即

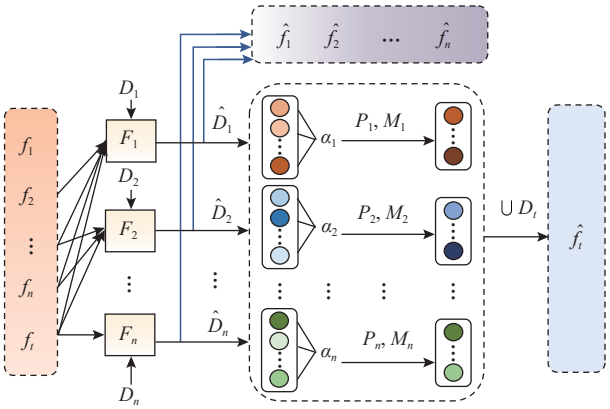
$$H(\boldsymbol{x})=\arg\max_{y}\sum_{i=1}^{n}[\![f_i(\boldsymbol{x})=y]\!]. \tag{15}$$

2.3 算法实施流程

TAEL方法首先检测漂移跨度$span$，判断漂移类型，然后在过滤阶段，设置非关键样本过滤器F，依次对历史样本块进行过滤操作，将剩余的历史关键样本用于训练更新历史基学习器，以提高其有效性；在扩充阶段，采用分块优先抽取策略，根据样本所属类别的规模占比设置抽样优先级，计算得到抽样概率，



(a) 突变型



(b) 渐变型

Fig. 3 Two-stage adaptive ensemble process

图 3 两阶段自适应集成过程

从历史关键样本块中抽取合适数量的样本子集来扩充当前样本块, 缓解了扩充后的类分布不均衡, 解决了当前基学习器欠拟合的问题. 算法 1 展示了 TAEL 方法的执行流程.

算法 1. 面向不同类型概念漂移的两阶段自适应集成算法.

输入: 历史数据块 $D_1, D_2, \cdots, D_n$, 当前数据块 D_t, 漂移跨度 $span$, 历史基学习器 $f_1, f_2, \cdots, f_n$, 当前基学习器 f_t, 非关键样本过滤器 F.

输出: 更新后的基学习器 $\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_n$, 和 $\hat{f}_t$.

① 获取 D_t 上每个类别的样本数 $\sum_{x=1}^{k} [\![y_{tx} = c]\!]$;

② if $span \leqslant \theta$

③ $F_i = f_t$;

④ else

⑤ $F_i = \sum_{p=i+1}^{n} \frac{w_p}{\sum_{q=i+1}^{n} w_q + w_t} f_p + \frac{w_t}{\sum_{q=i+1}^{n} w_q + w_t} f_t$;

⑥ end if

⑦ for $i = 1 : n$

⑧ for $j = 1 : k$

⑨ if $F_i(\boldsymbol{x}_{ij}) \neq y_{ij}$

⑩ 从 D_i 中删除样本 $\boldsymbol{x}_{ij}$;

⑪ else

⑫ 根据式(10)计算 $\boldsymbol{x}_{ij}$ 的抽样优先级 α_{ij};

⑬ end if

⑭ end for

⑮ 根据式(11)由抽样优先级 α_i 计算抽样概率 P_i;

⑯ 得到过滤后的历史关键数据块 $\hat{D}_i$;

⑰ if $span \leqslant \theta$

⑱ 更新历史基学习器 $\hat{f}_i \leftarrow train(\hat{D}_i \cup D_t)$;

⑲ else

⑳ 更新历史基学习器 $\hat{f}_i \leftarrow train(\hat{D}_i)$;

㉑ end if

㉒ end for

㉓ 获取总抽样规模 $M = \lambda \times \frac{span}{span+1} \times \sum_{i=1}^{n} z_i$;

㉔ for $i = 1 : n$

㉕ 根据式(9)计算每个 $\hat{D}_i$ 上的抽样规模 M_i;

㉖ 按照抽样概率 P_i 从 $\hat{D}_i$ 中抽取大小为 M_i 的 $\widetilde{D}_i = Sampling(\hat{D}_i, M_i, P_i)$;

㉗ end for

㉘ $\hat{D}_t = D_t \cup \widetilde{D}_1 \cup \widetilde{D}_2 \cup \cdots \cup \widetilde{D}_n$;

㉙ 更新当前基学习器 $\hat{f}_t \leftarrow train(\hat{D}_t)$;

㉚ 根据式(15)将最新更新的基学习器参与集成;

㉛ 在 D_{t+1} 上进行测试, 得到实时精度.

2.4 模型复杂度分析

TAEL 的计算成本主要集中在漂移类型检测、样本过滤、样本扩充和基学习器更新这 4 个阶段, 本文将依次对每个阶段进行时间复杂度分析.

1)漂移类型检测. 预测一个数据块中样本的时间复杂度为 $O(k)$, 其中 k 为数据块的样本数, 那么 L 个历史基学习器在当前数据块上的预测时间复杂度为 $O(Lk)$, 计算后序数据分布稳定程度的时间复杂度为 $O(L)$. 因此, 漂移类型检测过程的时间复杂度为 $O(Lk) + O(L) = O(Lk)$.

2)样本过滤. 训练一个 SVM 分类器的时间复杂度为 $O(p^2)$, 其中 p 为样本数, 因此, 在大小为 k 的数据块上训练基学习器的时间复杂度为 $O(k^2)$. 当发生突变型概念漂移时, 使用直接式过滤器, 当前基学习器对所有历史数据的预测时间复杂度为 $O(sk)$, 其中 s 为基学习器的最大存储容量. 该过程的时间复杂度

为 $O(k^2)$(一般地，$s<k$).

当发生渐变型概念漂移时，采用叠加式过滤器，根据历史和最新基学习器在当前数据块上的预测结果得到权值的时间复杂度为 $O((s+1)k)$. 对所有历史数据块依次执行基学习器的加权预测，总的时间复杂度为 $O(s(s+1)k)$. 因此，样本过滤过程的时间复杂度为 $O(k^2)+O((s+1)k)+O(s(s+1)k)=O(s^2k)$.

3)样本扩充. 计算所有历史数据块的抽样规模 M_i 的时间复杂度为 $O(s)$. 计算每个历史数据块中样本的抽样优先级 α 的时间复杂度为 $O(sk)$，计算每个样本的抽样概率 P 的时间复杂度为 $O(sk)$. 对于每个历史数据块，根据抽样概率 P 在当前数据块随机抽取 M_i 个样本的时间复杂度为 $O(sk)$. 该过程的时间复杂度为 $O(s)+3O(sk)=O(sk)$.

4)更新基学习器. 训练 $s+1$ 个基学习器的时间复杂度为 $O((s+1)k^2)$. 替换掉最差基学习器的时间复杂度 $O((s+1)k)$，整个过程的时间复杂度为 $O((s+1)(k^2+k))=O(sk^2)$.

3 实验分析

为验证本文提出的 TAEL 方法的有效性，本文在具有不同类型概念漂移的标准数据集和真实数据集上进行实验，并从精度、鲁棒性以及收敛性这 3 个方面进行评价. 实验平台为 Windows10 操作系统，CPU 为酷睿 i7-3，2 GHz 内核，内存为 8 GB，本方法采用 MATLAB R2018a 编写和运行.

3.1 实验数据

为了检验方法对不同类型概念漂移的处理能力，本文使用大规模在线分析平台 MOA[30] 中的流数据生成器产生了 6 个具有突变式、渐进式以及增量式的概念漂移数据集. 除此之外，本文还选取了 4 个真实数据集. 具体的数据集信息如表 1 所示.

Table 1 Datasets Information

表 1 数据集信息

分类	数据集	实例数	维度	类别数量	漂移类型	漂移数量	漂移位点
合成数据集	Sea	100×10^3	3	2	渐进式	3	25×10^3, 50×10^3, 75×10^3
	Hyperplane	100×10^3	10	2	增量式	-	-
	RBFBlips	100×10^3	20	4	突变式	3	25×10^3, 50×10^3, 75×10^3
	LED_abrupt	100×10^3	24	10	突变式	1	50×10^3
	LED_gradual	100×10^3	24	10	渐进式	3	25×10^3, 50×10^3, 75×10^3
	Tree	100×10^3	30	10	突变式	3	25×10^3, 50×10^3, 75×10^3
真实数据集	Electricity	45.3×10^3	6	2	-	-	-
	Kddcup99	494×10^3	41	23	-	-	-
	Covertype	581×10^3	54	7	-	-	-
	Weather	95.1×10^3	9	3	-	-	-

注：“-”表示未知.

3.2 评价指标

为衡量 TAEL 方法的性能，本节从模型的精度、鲁棒性及收敛性 3 方面进行了分析.

1) 平均实时精度(average real-time accuracy, *Avgracc*)表示模型在每个时间步的实时精度的平均值，反映模型的实时性能.

$$Avgracc=\frac{1}{T}\sum_{t=1}^{T}\frac{n_t}{|D_t|},\tag{16}$$

其中 n_t 代表时间步 t 内正确分类的样本数，$|D_t|$表示样本块大小，T 表示总的时间步数. 平均实时精度越高说明模型分类性能越好.

2)累积精度(cumulative accuracy, *Cumacc*)表示模型在当前时刻的累积预测正确样本数和总样本数的比值，反映模型从开始到当前时刻的整体性能.

$$Cumacc=\frac{\sum_{i=1}^{T_t}n_i}{\sum_{j=1}^{T_t}|D_j|},\tag{17}$$

其中 T_t 表示当前累积的时间步数.

3)鲁棒性(robustness, R)[31] 表示模型的稳定性和泛化性能. 本文在平均实时精度上分析了不同方法的鲁棒性，定义为：

$$R(Dataset)=\frac{racc(Dataset)}{\min racc(Dataset)},\tag{18}$$

其中 $racc(Dataset)$ 表示某算法在数据集 *Dataset* 上的平均实时精度，$\min racc(Dataset)$ 表示在数据集 *Dataset* 上所有算法中的最小平均实时精度.

某算法的整体鲁棒性值为该算法在所有数据集上的鲁棒性的总和. 鲁棒性值越大说明算法越稳定，面对数据中存在的干扰也能保持较好的性能.

4)收敛速度(recovery speed under accuracy, *RSA*)表示模型从概念漂移位点起实时精度恢复到稳定所需要的时间步数 *step* 与收敛位点后 K 个位点平均错误率 *avge* 的乘积：

$$RSA=step\times avge.\tag{19}$$

如果一个位点的性能表现和其后续 K 个参照位点的平均性能表现的差异小于阈值 γ(当前波动程度较小)，同时 K 个参照位点的前半部分和后半部分的

平均性能表现的差异小于$\frac{\gamma}{2}$(整体波动程度趋近于稳定)，那么该位点为收敛位点：

$$\left|acc_t-\frac{\sum_{j=1}^{K}acc_{t+j}}{K}\right|<\gamma \text{ 且 } \frac{2}{K}\left|\sum_{j=1}^{\frac{K}{2}}acc_{t+j}-\sum_{k=\frac{K}{2}+1}^{K}acc_{t+k}\right|<\frac{\gamma}{2}. \tag{20}$$

3.3 参数设置

本节对实验模型中的相关参数进行4点讨论：

1)数据块大小k. 过大的数据块中可能包含概念漂移，从而影响模型的分类效果；过小的数据块中可能无法包含足够多的样本特征，从而导致训练的基学习器稳定性较差. 因此，本文统一设置$k=500$.

2)漂移稳定性参数δ和漂移类型参数θ. 考虑到流数据本身的复杂性以及概念漂移类型的多样性，本文设置$\delta=0.01$，$\theta=1$.

3)样本规模因子λ. 样本规模控制了整体抽样的数量，直接影响了当前基学习器的训练，从而可能会对整体的模型性能造成影响. 因此，本文选取$\lambda\in\{0.2,0.4,0.6,0.8\}$进行讨论，得到了在不同$\lambda$下的分类性能，并使用最优样本规模因子与对比方法进行比较.

4)基学习器f. 本文选择LIBSVM来构建“同质”基学习器，核参数采用默认值$g=1/v$，(v为数据特征维度)，惩罚因子设置为$C=10$.

3.4 实验结果与分析

为评估TAEL的性能，本文选取DWCDS[17]，HBP[32]，Resnet[33]，Highway[34]以及原始深度神经网络(DNN)在精度、鲁棒性和收敛性3个方面进行对比实验和结果分析.

3.4.1 模型精度结果和分析

本节首先分析了在不同样本规模因子λ下集成模型的表现性能. 表2展示了TAEL方法在不同λ下的平均实时精度. 从表2可以看出当$\lambda=0.4$和$\lambda=0.6$时的平均实时精度值较高，这也反映了λ会在一定程度上影响当前基学习器的性能，进而影响整个集成模型的实时精度. 分析其原因可能是当λ取值较大时扩充的历史样本数太多，此时的关键信息冗余，训练得到的基学习器效果较差；当λ取值较小时扩充的样本数太少，可能丢弃潜在的可用数据，导致训练得到的基学习器处于欠拟合状态. 因此，本文选择适中的扩充规模，又因实验结果中$\lambda=0.4$时的平均实时精度大于$\lambda=0.6$时的平均实时精度，最终选择$\lambda=0.4$的情况下与其他方法进行对比分析.

Table 2 Average Real-Time Accuracy Under Different λ

表2 不同λ下平均实时精度

数据集	平均实时精度(排名)			
	λ=0.2	λ=0.4	λ=0.6	λ=0.8
Sea	**0.838 9 (1)**	0.837 8 (3)	0.837 8 (3)	0.837 8 (3)
Hyperplane	0.910 9 (4)	0.911 0 (2.5)	**0.911 1 (1)**	0.911 0 (2.5)
RBFBlips	**0.954 9 (2.5)**	**0.954 9 (2.5)**	**0.954 9 (2.5)**	**0.954 9 (2.5)**
LED_abrupt	0.622 8 (3)	**0.622 9 (2)**	**0.622 9 (2)**	**0.622 9 (2)**
LED_gradual	0.620 5 (3)	**0.622 6 (1)**	0.619 9 (4)	0.621 3 (2)
Tree	**0.667 1 (1)**	0.666 9 (2)	0.666 0 (3)	0.665 6 (4)
Electricity	0.720 5 (2)	0.719 3 (3)	**0.721 1 (1)**	0.719 0 (4)
Kddcup99	0.938 4 (4)	0.944 9 (2)	**0.945 5 (1)**	0.944 8 (3)
Covertype	0.752 0 (2)	**0.752 6 (1)**	0.751 7 (3.5)	0.751 7 (3.5)
Weather	0.896 9 (3.5)	**0.897 0 (1.5)**	**0.897 0 (1.5)**	0.896 9 (3.5)
平均排名	2.6	**2.05**	2.25	3.0

注：黑体数字表示最高平均实时精度及其排名.

表3展示了不同方法在所有数据集上的平均实时精度及其综合排名. 由表3看出，在合成数据集上，TAEL的实时精度最好；在真实数据集上，TAEL的实时精度排名也都位于前列. TAEL在真实数据集上排名较低的原因可能在于数据集中概念漂移的出现较为密集，而TAEL利用数据块进行处理的方式可能会漏检，导致无法对基学习器进行及时地更新，从而使整个集成模型的性能下降. 在整体排名上TAEL的排名最高，说明了该方法能够提高集成模型的有效性，有较好处理不同类型概念漂移的能力.

图4为TAEL和各个对比方法在所有数据集上的累积精度，表4为TAEL和各个对比方法的最终累积精度和综合排名. 由图4和表4可知，在标准数据集上TAEL的累积精度最高，在真实数据集上TAEL的累积精度也有较好的排名，分析其原因是该方法针对漂移类型对数据块逐一处理的策略能够使模型对不同类型的概念漂移做出及时响应，保持较高的精度.

本文使用非参数检验方法Friedman-Test[35]对TAEL与对比方法相比较的性能优势进行统计检验. 对于给定的K(K=9)种方法和N(N=10)个数据集，令r_i^j为第j个方法在第i个数据集上的秩，则第j个算法的秩和平均为

$$R_j=\frac{1}{N}\sum_{i=1}^{N}r_i^j. \tag{21}$$

Table 3 Average Real-Time Accuracy of Different Methods on Each Dataset

表 3 不同方法在各数据集上的平均实时精度

数据集	平均实时精度（排名）								
	DWCDS	DNN-2	DNN-4	DNN-8	DNN-16	HBP	Highway	Resnet	TAEL
Sea	0.749 9 (4)	0.708 1 (9)	0.715 5 (8)	0.749 5 (5)	0.744 1 (7)	0.777 1 (2)	0.768 4 (3)	0.744 8 (6)	**0.837 8 (1)**
Hyperplane	0.681 2 (9)	0.860 0 (5)	0.857 8 (6)	0.848 7 (7)	0.722 7 (8)	0.869 2 (3)	0.884 1 (2)	0.863 7 (4)	**0.911 0 (1)**
RBFBlips	0.821 4 (8)	0.825 6 (7)	0.871 6 (2)	0.865 5 (3)	0.471 8 (9)	0.835 0 (5)	0.848 2 (4)	0.830 0 (6)	**0.954 9 (1)**
LED_abrupt	0.370 0 (8)	0.586 8 (3)	0.580 9 (4)	0.531 1 (7)	0.278 4 (9)	0.569 2 (6)	0.589 3 (2)	0.579 6 (5)	**0.622 9 (1)**
LED_gradual	0.380 4 (8)	0.577 3 (4)	0.589 8 (2)	0.535 0 (7)	0.303 1 (9)	0.565 0 (6)	0.583 9 (3)	0.570 0 (5)	**0.619 9 (1)**
Tree	0.555 8 (2)	0.194 8 (6)	0.205 7 (3)	0.133 8 (8)	0.114 1 (9)	0.143 2 (7)	0.203 6 (4)	0.199 2 (5)	**0.666 9 (1)**
Electricity	**0.734 6 (1)**	0.622 8 (6)	0.623 1 (5)	0.563 5 (8)	0.515 4 (9)	0.567 6 (7)	0.631 7 (4)	0.634 3 (3)	0.719 3 (2)
Kddcup99	**0.982 9 (1)**	0.879 6 (3)	0.718 6 (6)	0.476 3 (8)	0.301 7 (9)	0.767 0 (4)	0.753 7 (5)	0.653 5 (7)	0.944 9 (2)
Covertype	**0.848 6 (1)**	0.525 1 (9)	0.573 9 (8)	0.624 3 (6)	0.626 9 (5)	0.646 5 (3)	0.635 4 (4)	0.618 3 (7)	0.752 6 (2)
Weather	**0.956 6 (1)**	0.847 8 (3)	0.805 0 (6)	0.805 7 (5)	0.804 3 (7)	0.813 9 (4)	0.781 3 (9)	0.803 4 (8)	0.897 0 (2)
平均排名	4.30	5.50	5.00	6.40	8.10	4.70	4.00	5.60	**1.40**

注：黑体数字表示最高平均实时精度及其排名.

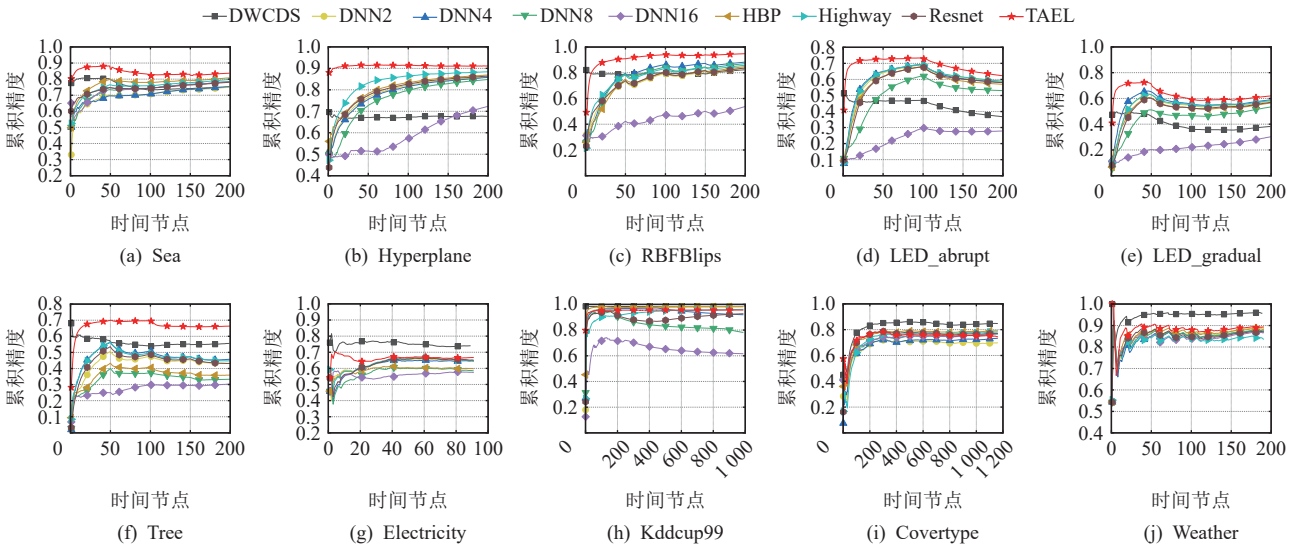


Fig. 4 Comparison of cumulative accuracy of different methods on each dataset

图 4 不同方法在各数据集上的累积精度比较

零假设 H_0 假定所有方法的性能是相同的. 在此前提下, 当 N 与 K 足够大时, Friedman 统计值 τ_F 服从第一自由度为 $K-1$、第二自由度为 $(K-1)(N-1)$ 的 F 分布:

$$\tau_F = \frac{(N-1)\tau_{\chi^2}}{N(K-1)-\tau_{\chi^2}}, \tag{22}$$

$$\tau_{\chi^2} = \frac{12N}{K(K+1)}\left[\sum_{j=1}^{K} R_j^2 - \frac{K(K+1)^2}{4}\right].$$

若计算得到的统计值大于某一显著性水平下 F 分布临界值, 则拒绝零假设 H_0, 表明各方法的秩和存在显著差异, 即测试方法性能存在显著差异; 反之则接受零假设 H_0, 所有方法的性能没有明显差异.

在 $\alpha=0.05$ 的情况下 F 分布临界值 $\tau_F^{0.05}(8,72)=2.069\,8$, 经计算可得在不同性能指标下的 Friedman 统计值 τ_F, 如表 5 所示. 从表 5 可以看出, 平均实时精度和最终累积精度下的 τ_F 统计值均大于临界值 $\tau_F^{0.05}(8,72)$, 拒绝零假设 H_0, 说明所有方法性能存在显著差异.

本文用 Bonferroni-Dunn 测试[36] 计算了所有方法的显著性差异, 用于比较 2 种方法之间是否存在显著差异. 若 2 种方法的秩和平均差值大于临界差, 则这 2 种方法的性能存在显著差异:

$$CD = q_\alpha\sqrt{\frac{K(K+1)}{6N}}, \tag{23}$$

其中当 $K=9$, $N=10$ 时, 可以查表得到 $q_{\alpha=0.05}=2.724$,

Table 4 Final Cumulative Accuracy of Different Methods on Each Dataset

表 4 不同方法在各数据集上的最终累积精度

数据集	最终累积精度（排名）								
	DWCDS	DNN-2	DNN-4	DNN-8	DNN-16	HBP	Highway	Resnet	TAEL
Sea	0.750 0(8)	0.749 5(9)	0.754 3(7)	0.786 1(4)	0.782 0(5)	0.808 3(2)	0.797 7(3)	0.780 3(6)	**0.837 0(1)**
Hyperplane	0.676 3(9)	0.860 0(5)	0.858 0(6)	0.848 3(7)	0.723 0(8)	0.869 1(3)	0.884 0(2)	0.863 6(4)	**0.911 0(1)**
RBFBlips	0.823 1(8)	0.834 5(7)	0.882 8(2)	0.870 8(3)	0.537 9(9)	0.847 6(5)	0.858 6(4)	0.837 4(6)	**0.948 1(1)**
LED_abrupt	0.368 1(8)	0.586 9(3)	0.580 3(4)	0.530 5(7)	0.278 6(9)	0.569 3(6)	0.589 3(2)	0.579 6(5)	**0.622 9(1)**
LED_gradual	0.382 1(8)	0.577 6(4)	0.589 8(2)	0.534 4(7)	0.303 2(9)	0.565 0(6)	0.584 3(3)	0.569 9(5)	**0.619 9(1)**
Tree	0.555 8(2)	0.432 9(5)	0.457 5(3)	0.333 0(8)	0.303 3(9)	0.359 1(7)	0.447 2(4)	0.431 0(6)	**0.663 6(1)**
Electricity	**0.740 4(1)**	0.643 4(6)	0.645 0(4)	0.584 0(8)	0.573 5(9)	0.596 9(7)	0.644 7(5)	0.650 2(3)	0.667 4(2)
Kddcup99	**0.983 3(1)**	0.983 2(2)	0.919 5(7)	0.781 3(8)	0.616 0(9)	0.982 3(3)	0.961 4(4)	0.927 6(6)	0.956 2(5)
Covertype	**0.848 1(1)**	0.698 3(9)	0.733 6(8)	0.767 6(6)	0.768 5(5)	0.791 9(2)	0.782 3(3)	0.770 9(4)	0.746 3(7)
Weather	**0.957 1(1)**	0.887 2(3)	0.874 3(6)	0.875 4(5)	0.866 4(8)	0.882 4(4)	0.836 2(9)	0.870 8(7)	0.893 3(2)
平均排名	4.70	5.30	4.90	6.30	8.00	4.50	3.90	5.20	**2.20**

注：黑体数字表示最高的最终累积精度及其排名.

Table 5 τ_F of Average Real-Time Accuracy and Final Cumulative Accuracy

表 5 平均实时精度和最终累积精度下的τ_F

评价指标	τ_F	$\tau_F^{0.05}(8,72)$
平均实时精度	7.226 0	2.069 8
最终累积精度	4.574 7	

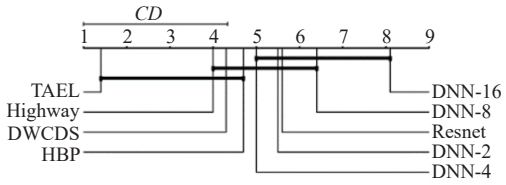
经计算得到显著性水平$\alpha=0.05$的情况$CD=3.336\,2$. 不同方法在平均实时精度和最终累积精度上的统计分析结果如图 5 所示，在图中将没有显著性差异的方法使用黑线连接起来. 结果表明，在统计意义上，TAEL 方法排名最好且具有明显的优势.

3.4.2 模型鲁棒性分析

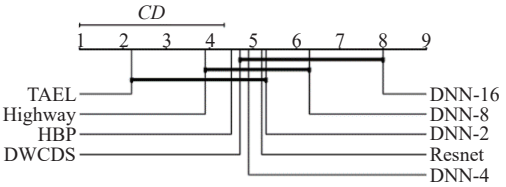
为了衡量各个方法的算法稳定性，本节计算每个方法在各个数据集上的鲁棒性，图 6 展示了计算结果. 图 6 中每个小矩形的面积代表的是算法在某种数据集上的鲁棒性值的大小，每一列上展示的数值代表算法在所有数据集上的鲁棒性值总和，即该算法的整体鲁棒性. 由图 6 可知，在大多数情况下，TAEL 的鲁棒性都能取得较好的排名，且整体鲁棒性最高，这说明该方法对数据的噪声和异常值具有更强的鲁棒性，能提高集成模型的整体泛化性能.

3.4.3 模型收敛性分析

为比较各个方法在概念漂移发生后的收敛性能，本节计算并分析了各个方法在 5 个合成数据集的概念漂移位点上的收敛速度. 在收敛位点的判定过程中，设定收敛判定阈值$\gamma=0.02$，参照位点个数$K=20$. 表 6 展示的为各个方法在数据集上的已知漂移位点上计算得到的收敛速度. 由于个别方法在漂移位点



(a) 平均实时精度



(b) 最终累积精度

Fig. 5 Analysis of critical difference in average real-time accuracy and final cumulative accuracy of different methods

图 5 不同方法在平均实时精度和最终累积精度上的显著性差异分析

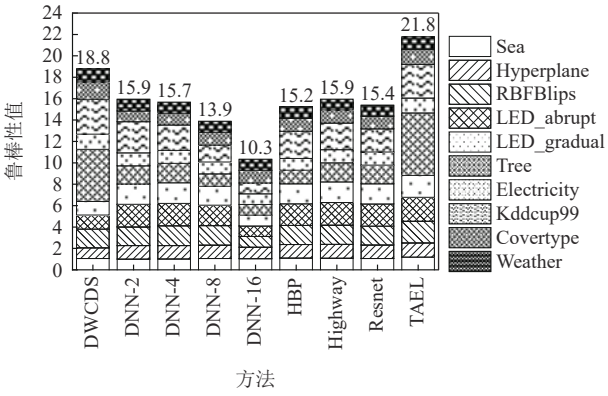


Fig. 6 Comparison of robustness of different methods

图 6 不同方法的鲁棒性比较

处精度保持平稳波动，因此，对该位点的收敛速度不做统计，用“-”进行表示. 从表 6 可以看出，TAEL 在多数情况下都具有较快的收敛速度，是因为该方法及时更新基学习器使其能尽快适应新的数据分布，集成有效性得到提高. 在整体排名中 TAEL 处于第一，说明该方法具有较快的收敛速度，收敛性能较好.

Table 6 Recovery Speed Under Accuracy of Different Methods on Each Dataset

表 6 不同方法在各数据集上的收敛速度

数据集	DWCDS	DNN-2	DNN-4	DNN-8	DNN-16
Sea	0.67/**0.25**/0.45	0.97/2.63/0.70	1.24/1.15/2.18	2.36/1.55/2.78	2.60/1.66/**0.20**
RBFBlips	0.56/0.85/0.16	1.17/1.56/0.41	0.38/0.61/0.22	0.56/0.94/0.21	-/-/-
LED_abrupt	**3.70**	9.77	14.92	19.85	17.94
LED_gradual	**2.20**/**1.95**/-	10.70/7.43/6.01	10.91/7.89/**3.58**	14.24/11.48/3.70	14.83/9.61/4.55
Tree	9.29/4.69/4.70	-/23.81/**0.87**	2.62/15.22/7.69	4.44/0.88/0.88	**0.89**/0.88/0.88
平均排名	3.54	5.85	4.85	5.54	5.77

数据集	HBP	Highway	Resnet	TAEL
Sea	0.77/0.51/1.82	2.76/0.49/1.81	0.57/0.56/0.73	**0.45**/1.97/1.50
RBFBlips	1.14/0.83/0.21	0.85/1.14/0.42	1.01/2.02/0.62	**0.03/0.29/0.01**
LED_abrupt	20.20	9.80	11.65	5.28
LED_gradual	13.12/10.86/7.42	9.66/6.68/5.37	12.80/7.06/5.47	8.25/5.34/3.71
Tree	3.56/**0.87**/1.75	-/17.56/1.75	-/21.49/1.75	3.85/3.92/3.85
平均排名	5.38	5.23	5.69	**3.15**

注：“-”表示对当前位点的收敛速度不进行统计；黑体数字表示最高收敛速度；LED_abrupt 包含 1 个漂移位点，收敛速度只有 1 个；其他数据集包含 3 个漂移位点，对应 3 个收敛速度.

4 结束语

针对概念漂移发生后，在线集成模型无法及时响应数据流的变化而导致泛化性能降低、收敛速度减慢的问题，本文提出一种面向不同类型概念漂移的两阶段自适应集成学习方法. 本文通过检测漂移跨度来确定漂移类型，并采用一种针对漂移类型进行自适应调整的两阶段样本处理机制. 在该机制中，一方面通过样本过滤策略过滤历史样本块中的非关键样本，使历史数据分布更接近当前最新数据分布，提高了基学习器的有效性；另一方面通过样本扩充策略为当前样本集补充合适数量的历史关键样本，解决了当前基学习器的欠拟合问题，同时缓解了扩充后的类别不平衡. 更新后的基学习器组成的集成模型的有效性得到了提高，对不同类型的概念漂移能做出更精准及时的响应. 在集成学习中，集成的多样性同样影响了集成模型的性能，在未来的工作中，将进一步研究针对不同漂移类型提升集成多样性的方法.

作者贡献声明：郭虎升负责思想提出、方法设计、论文写作及修改；张洋负责论文写作、代码实现、数据测试及论文修改；王文剑负责写作指导、修改审定.

参 考 文 献

[1] Rutkowski L, Jaworski M, Duda P, et al. Basic concepts of data stream mining[J]. Stream Data Mining:Algorithms and Their Probabilistic Properties, 2020, 56: 13−33

[2] Zhai Tingting, Gao Yang, Zhu Junwu. Survey of online learning algorithms for streaming data classification[J]. Journal of Software, 2020, 31(4): 912−931 (in Chinese)
(翟婷婷, 高阳, 朱俊武. 面向流数据分类的在线学习综述[J]. 软件学报, 2020, 31(4): 912−931)

[3] Wang Tao, Li Zoujun, Yan Yuejin, et al. A survey of classification of data stream[J]. Journal of Computer Research and Development, 2007, 44(11): 1809−1815 (in Chinese)
(王涛, 李舟军, 颜跃进, 等. 数据流挖掘分类技术综述[J]. 计算机研究与发展, 2007, 44(11): 1809−1815)

[4] Du Hangyuan, Wang Wenjian, Bai Liang. A novel evolving data stream clustering method based on optimization model[J]. SCIENTIA SINICA Informationis, 2017, 47(11): 1464−1482 (in Chinese)
(杜航原, 王文剑, 白亮. 一种基于优化模型的演化数据流聚类方法[J]. 中国科学: 信息科学, 2017, 47(11): 1464−1482)

[5] Krempl G, Žliobaite I, Brzeziński D, et al. Open challenges for data

stream mining research[J]. ACM SIGKDD Explorations Newsletter, 2014, 16(1): 1−10

[6] Wen Yimin, Liu Shuai, Miao Yuqing, et al. Survey on semi-supervised classification of data streams with concept drifts[J]. Journal of Software, 2022, 33(4): 1287−1314 (in Chinese)
(文益民, 刘帅, 缪裕青, 等. 概念漂移数据流半监督分类综述[J]. 软件学报, 2022, 33(4): 1287−1314)

[7] Lu Jie, Liu Anjin, Dong Fan, et al. Learning under concept drift: A review[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(12): 2346−2363

[8] Guo Husheng, Zhang Aijuan, Wang Wenjian. Concept drift detection method based on online performance test[J]. Journal of Software, 2020, 31(4): 932−947 (in Chinese)
(郭虎升, 张爱娟, 王文剑. 基于在线性能测试的概念漂移检测方法[J]. 软件学报, 2020, 31(4): 932−947)

[9] Lu Ning, Zhang Guangquan, Lu Jie. Concept drift detection via competence models[J]. Artificial Intelligence, 2014, 209: 11−28

[10] Krawczyk B, Minku L L, Gama J, et al. Ensemble learning for data stream analysis: A survey[J]. Information Fusion, 2017, 37: 132−156

[11] Liang Bin, Li Guanghui, Dai Chenglong. G-mean weighted classification method for imbalanced data stream with concept drift[J]. Journal of Computer Research and Development, 2022, 59(12): 2844−2857 (in Chinese)
(梁斌, 李光辉, 代成龙. 面向概念漂移且不平衡数据流的 G-mean 加权分类方法[J]. 计算机研究与发展, 2022, 59(12): 2844−2857)

[12] Gomes H M, Barddal J P, Enembreck F, et al. A survey on ensemble learning for data stream classification[J]. ACM Computing Surveys, 2017, 50(2): 1−36

[13] Webb G I, Hyde R, Cao Hong, et al. Characterizing concept drift[J]. Data Mining and Knowledge Discovery, 2016, 30(4): 964−994

[14] Bifet A, Gavalda R. Learning from time-changing data with adaptive windowing [C]//Proc of the 7th SIAM Int Conf on Data Mining. Philadelphia, PA: SIAM, 2007: 443−448

[15] Gama J, Medas P, Castillo G, et al. Learning with drift detection [C]//Proc of the 17th Brazilian Symp on Artificial Intelligence. Berlin: Springer, 2004: 286−295

[16] Nishida K, Yamauchi K. Detecting concept drift using statistical testing [C]//Proc of the 10th Int Conf on Discovery Science. Berlin: Springer, 2007: 264−269

[17] Zhu Qun, Hu Xuegang, Zhang Yuhong, et al. A double-window-based classification algorithm for concept drifting data streams [C]//Proc of 2010 IEEE Int Conf on Granular Computing. Piscataway, NJ: IEEE, 2010: 639−644

[18] Guo Husheng, Ren Qiaoyan, Wang Wenjian. Concept drift class detection based on time window[J]. Journal of Computer Research and Development, 2022, 59(1): 127−143 (in Chinese)
(郭虎升, 任巧燕, 王文剑. 基于时序窗口的概念漂移类别检测[J]. 计算机研究与发展, 2022, 59(1): 127−143)

[19] Guo Husheng, Li Hai, Ren Qiaoyan, et al. Concept drift type identification based on multi-sliding windows[J]. Information Sciences, 2022, 585: 1−23

[20] Sidhu P, Bhatia M P S. An online ensembles approach for handling concept drift in data streams: Diversified online ensembles detection[J]. International Journal of Machine Learning and Cybernetics, 2015, 6(6): 883−909

[21] Minku L L, White A P, Yao Xin. The impact of diversity on online ensemble learning in the presence of concept drift[J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 22(5): 730−742

[22] Shan Jicheng, Zhang Hang, Liu Weike, et al. Online active learning ensemble framework for drifted data streams[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 30(2): 486−498

[23] Sun Yu, Tang Ke, Minku L L, et al. Online ensemble learning of data streams with gradually evolved classes[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(6): 1532−1545

[24] Street W N, Kim Y S. A streaming ensemble algorithm (SEA) for large-scale classification [C]//Proc of the 7th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2001: 377-382

[25] Lu Yang, Cheung Y M, Tang Yuanyan. Dynamic weighted majority for incremental learning of imbalanced data streams with concept drift [C]//Proc of the 26th Int Joint Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2017: 2393−2399

[26] Lu Yang, Cheung Y M, Tang Yuanyan. Adaptive chunk-based dynamic weighted majority for imbalanced data streams with concept drift[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 31(8): 2764−2778

[27] Ren Siqi, Zhu Wen, Liao Bo, et al. Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning[J]. Knowledge-Based Systems, 2019, 163: 705−722

[28] Li Zeng, Huang Wenchao, Xiong Yan, et al. Incremental learning imbalanced data streams with concept drift: The dynamic updated ensemble algorithm[J]. Knowledge-Based Systems, 2020, 195: 105694

[29] Guo Husheng, Zhang Shuai, Wang Wenjian. Selective ensemble-based online adaptive deep neural networks for streaming data with concept drift[J]. Neural Networks, 2021, 142: 437−456

[30] Bifet A, Holmes G, Pfahringer B, et al. MOA: Massive online analysis, a framework for stream classification and clustering [C]//Proc of the 1st Workshop on Applications of Pattern Analysis. New York: PMLR, 2010: 44−50

[31] Zhao Peng, Zhou Zhihua. Learning from distribution-changing data streams via decision tree model reuse [J]. SCIENTIA SINICA Informationis, 2021, 51(1): 1−12 (in Chinese)
(赵鹏, 周志华. 基于决策树模型重用的分布变化流数据学习 [J]. 中国科学: 信息科学, 2021, 51(1): 1−12)

[32] Sahoo D, Pham Q, Lu Jing, et al. Online deep learning: Learning deep neural networks on the fly [C]//Proc of the 27th Int Joint Conf on Artificial Intelligence. Amsterdam: Elsevier, 2018: 2660−2666

[33] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition [C]//Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 770−778

[34] Srivastava R K, Greff K, Schmidhuber J. Training very deep networks

[C]//Proc of the 28th Int Conf on Neural Information Processing Systems. Cambridge, MA: MIT, 2015: 2377−2385

[35] Pereira D G, Afonso A, Medeiros F M. Overview of Friedman's test and post-hoc analysis[J]. Communications in Statistics-Simulation and Computation, 2015, 44(10): 2636−2653

[36] Demšar J. Statistical comparisons of classifiers over multiple data sets[J]. The Journal of Machine Learning Research, 2006, 7: 1−30

Guo Husheng, born in 1986. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include data mining, machine learning, and computational intelligence.

郭虎升, 1986 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为数据挖掘、机器学习、计算智能.

Zhang Yang, born in 1999. Master. Her main research interests include streaming data mining and online machine learning.

张　洋, 1999 年生. 硕士. 主要研究方向为流数据挖掘、在线机器学习.

Wang Wenjian, born in 1968. PhD, professor, PhD supervisor. Distinguished member of CCF. Her main research interests include machine learning, data mining, and computational intelligence.

王文剑, 1968 年生. 博士, 教授, 博士生导师. CCF 杰出会员. 主要研究方向为机器学习、数据挖掘、计算智能.