计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202220835
61(1): 172−183, 2024

基于自适应深度集成网络的概念漂移收敛方法

郭虎升[1,2]　孙　妮[1]　王嘉豪[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)

Concept Drift Convergence Method Based on Adaptive Deep Ensemble Networks

Guo Husheng[1,2], Sun Ni[1], Wang Jiahao[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　Concept drift is an important and challenging problem in streaming data mining field. However, most existing methods can only deal with linear or simple nonlinear mappings. In spite of the ability of fitting nonlinear functions, neural network models have difficulty in adjusting dynamically according to the changing data streaming because only one sample or one batch of samples is available at a time for model training in the context of streaming data mining task. In order to solve above problem, the thought of gradient boosting algorithm is introduced to solve the problem of streaming data mining task with concept drift and a concept drift convergence method based on adaptive deep ensemble networks (CD_ADEN) is proposed. The proposed model combines several shallow neural networks as base leaner, and subsequent base learner corrects the output of precedent base learner to make the final output achieve high real-time generalization performance. Besides, because of the high convergence speed of shallow neural network, the proposed model will quickly recover from accuracy decrease caused by concept drift. The experimental results on multiple datasets show that the average real-time accuracy of the proposed CD_ADEN method is significantly improved compared with the comparative methods, the average real-time accuracy is improved by 1%−5%, and the average ordinal value ranks first in the comparison of several algorithms. It shows that the proposed model can correct the error of the pre-order output, and the learning model can quickly recover from the accuracy drop caused by concept drift, which improves the real-time generalization performance of the online learning model.

Key words　streaming data; concept drift; gradient boosting; deep learning; quick adaptation

摘　要　概念漂移是流数据挖掘领域中的一个重要且具有挑战性的难题.然而,目前的方法大多仅能够处理线性或简单的非线性映射,深度神经网络虽然有较强的非线性拟合能力,但在流数据挖掘任务中,每次只能在新得到的1个或一批样本上进行训练,学习模型难以实时调整以适应动态变化的数据流.为解决上述问题,将梯度提升算法的纠错思想引入含概念漂移的流数据挖掘任务之中,提出了一种基于自适应深度集成网络的概念漂移收敛方法(concept drift convergence method based on adaptive deep ensemble networks, CD_ADEN).该模型集成多个浅层神经网络作为基学习器,后序基学习器在前序基学习器输出的基础上不

收稿日期:2022-09-28;**修回日期**:2023-04-06

基金项目:国家自然科学基金项目(62276157, U21A20513, 62076154);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) and the Key Research and Development Program of Shanxi Province (202202020101003).

通信作者:王文剑(wjwang@sxu.edu.cn)

断纠错，具有较高的实时泛化性能. 此外，由于浅层神经网络有较快的收敛速度，因此所提出的模型能够较快地从概念漂移造成的精度下降中恢复. 多个数据集上的实验结果表明，所提出的 CD_ADEN 方法平均实时精度有明显提高，相较于对比方法，平均实时精度有 1%~5% 的提升，且平均序值在 7 种典型的对比算法中排名第一. 说明所提出的方法能够对前序输出进行纠错，且学习模型能够快速地从概念漂移造成的精度下降中恢复，提升了在线学习模型的实时泛化性能.

关键词 流数据；概念漂移；梯度提升；深度学习；快速适应

中图法分类号 TP18

随着大数据时代来临，传感器数据、在线交易数据、社交媒体数据、用户在线行为数据等源源不断地产生. 不同于传统的静态数据，这些数据具有实时性、无限性、不可再生性等特点，称之为流数据. 近年来，互联网产业与人工智能技术迅猛发展，流数据挖掘的研究价值正变得愈来愈重要.

在流数据挖掘任务中，学习器往往需要进行动态调整以适应数据的实时变化. 例如，对于推荐系统，用户的喜好可能随着时间改变，导致数据分布随着时间变化，不同时刻的数据不再满足独立同分布假设，数据分布的这种变化将导致后验概率的变化，因此先前训练得到的学习模型不再适用于新分布的数据，流数据的这种分布变化被称为概念漂移[1-3].

目前许多在线学习模型都基于集成学习技术，利用多个学习器的加权或投票来取得比单个学习器更好的性能[4-5]. 同时，根据学习模型是否主动对概念漂移进行检测，在线学习方法可分为基于漂移检测的主动在线集成方法和自适应的被动在线集成方法. 基于漂移检测的主动在线集成方法大多通过模型性能指标（例如准确率）的变化来检测概念漂移，当模型性能发生大幅度波动时，说明流数据可能发生概念漂移，此时模型会用一个新的学习器来替换效果最差的旧学习器[6-10]. 然而，这种主动在线集成方法易受噪声或数据块大小的影响，造成学习器性能的大幅度波动，并且可能不能对概念漂移正确判断，导致模型性能不佳. 自适应型的被动在线集成方法[11-14]不直接检测概念漂移位点，而是根据流数据的变化情况动态地调整模型权重. 然而，这种方法难以利用历史信息去调节每个学习器权重以达到适应性和稳定性之间的平衡，并且不能很好地处理概念漂移. 此外，大多数在线学习方法都是基于线性分类器或者简单的非线性分类器（例如决策树、支持向量机、聚类等）[15-17]，这些分类器表征能力较弱，且难以处理较复杂的流数据挖掘任务.

近年来，深度学习技术在图像分类[18]、目标检测[19]、机器翻译[20]等领域取得了巨大成功. 与其他方法相比，深度学习技术拥有强大的表征能力，可以应用于许多复杂任务. 传统的深度学习方法使用大规模数据集对模型进行训练，并通过验证集来选择合适的模型架构（例如网络层的数量、每层神经元数量等），使模型能够有较好的泛化性能. 然而，在流数据挖掘任务中，模型每次只能在新进入的 1 个或者一小部分数据上训练和更新模型，无法利用验证集来进行模型架构选择，这就要求模型必须能够实时地动态调整以适应不断变化的数据流.

在线深度学习方法利用深度神经网络强大的表征能力去处理流数据以得到更好的效果. 尽管可以利用反向传播算法[21]对流数据直接优化损失函数，但这种方式仍存在一些问题. 首先，在流数据挖掘任务中，模型单次只能处理 1 个或一小批数据，这就要求模型要有很灵活的结构，能够在动态变化的环境中进行实时调整. 然而，不同层次的深度网络模型往往有不同的收敛速度和表征能力以适用于不同场景. 对于较浅层的深度网络，模型收敛速度较快，能够快速地达到较好的性能，但表征能力较差，难以处理复杂分布的问题，较适用于概念漂移刚发生时的场景，以使模型性能在漂移发生后快速回升. 对于深度网络，收敛速度往往比浅层网络要缓慢，但具有比浅层神经网络更强大的表征能力，因此更适用于数据流趋于稳定之后的场景. 此外，深度网络所面临的梯度消失等问题，也会影响其在流数据挖掘任务上的性能.

本文将梯度提升思想[22]引入含概念漂移的流数据挖掘问题中，提出了一种基于自适应深度集成网络的概念漂移收敛方法（concept drift convergence method based on adaptive deep ensemble networks, CD_ADEN）. CD_ADEN 方法将若干浅层网络作为基学习器进行集成，后序基学习器在前序基学习器输出基础上不断进行纠错，以提升学习模型在含概念漂移数据流分类任务中的实时泛化性能. 此外，由于每个基学习

器均为浅层网络, 有较快的收敛速度, 在概念漂移发生后, 学习模型能够较快地从漂移造成的性能下降中恢复. 实验结果表明, 对于不同的评价指标, 该方法平均排名均为第一. 在 Sea, RBFBlips, Tree, Covertype 等数据集上, 最终累积精度相对于对比方法分别提高了 1.63%, 0.73%, 3.48%, 2.2%. 本文的主要贡献包括 2 个方面:

1) 通过在前序输出的基础上对损失函数进行优化来替代学习损失函数梯度的方法, 提升学习模型对前序学习器输出的纠错性能.

2) 将梯度提升思想引入含概念漂移的流数据挖掘任务中, 后序基学习器在前序学习器输出的基础上不断纠错, 为解决含概念漂移的流数据挖掘问题提供了一个新思路.

1 相关工作

随着互联网技术的迅速发展, 越来越多的机器学习任务的数据以流数据的形式呈现, 概念漂移作为流数据挖掘领域中的一个重要难题, 受到越来越多的关注. 针对含概念漂移的流数据挖掘任务, 目前已经有较多研究. 根据是否对概念漂移的发生进行检测, 大致可分为基于主动漂移检测的方法和被动漂移适应的方法[23].

基于主动漂移检测的方法将概念漂移检测融入模型中, 通过对学习模型的性能指标进行监测来判断是否发生概念漂移. 一旦检测到模型的性能指标出现大幅度下降, 就认为数据流中发生了概念漂移, 并对模型做出相应调整. 典型的如: 滑动窗口方法[24]通过窗口不断向前滑动, 利用标准差等指标判断是否发生概念漂移, 若发生了概念漂移, 就将已失效的样本丢弃以适应新的数据分布; 增量贝叶斯方法[25]首先通过处理历史数据得到先验概率, 再结合半监督方法计算后验概率来检测概念漂移; 基于滑动窗口的 ACDWM 算法[4]通过设置 1 个滑动窗口来保存最新的数据, 当任意 2 个子窗口中数据的平均值的差别大于一定阈值的时候, 就认为发生了概念漂移, 就会丢弃过时的数据; CD-TW 算法[26-27]构建 2 个相邻的时序窗口, 通过比较 2 个窗口所对应数据的分布变化情况来检测概念漂移节点. 文献 [4, 24−27] 所述的这些方法虽然能在一定程度上处理概念漂移问题, 但仍存在概念漂移误报、概念漂移检测延迟和处理概念漂移类型单一等问题.

被动漂移适应的方法并不直接对概念漂移进行检测, 而是通过逐渐更新学习器以及根据每个基学习器的实时表现动态更新权重来适应概念漂移. 典型的如: SEA 算法[28]在每个数据块上都训练一个分类器, 然后将其加入集成学习模型中, 当集成学习模型中的基学习器的数量达到上限时, 替换表现最差的基学习器; 动态加权投票算法 DWM[29]根据每个基学习器的实时表现来动态地调整权重, 在更新权重的同时, 将权重小于某一阈值的基学习器移除; CondorForest 算法[30]通过决策树模型重用机制动态调整模型中的基学习器; 重用模型 Learn++.NSE 算法[31]是一种基于 AdaBoost 的动态加权算法, 该算法根据每个基学习器与当前数据分布的一致性来对基学习器进行加权, 并且在训练阶段通过一种遗忘机制来处理已过时的基学习器, 在不同类型的概念漂移都取得了较好的效果. 文献 [28−29, 31] 所述的这些方法在对基学习器进行加权时无法有效利用基学习器的历史信息达到适应性与稳定性之间的平衡, 往往不能很好地处理概念漂移.

尽管以上方法可以很好地解决概念漂移问题, 但多数算法都是通过在线凸优化方法来训练的一些简单模型, 对于复杂非线性任务往往难以处理. 而深度网络模型能够很好地拟合复杂的非线性函数, 但往往需要对大量数据进行批量训练. 目前用于专门处理流数据的深度网络模型仍比较少, 根据是否对网络架构进行动态调整可分为动态调整的在线深度网络和结构稳定的在线深度网络. 动态调整的在线深度网络不断对网络参数进行调整以适应最新的数据分布, 并根据网络实时性能来决定调整速率, 典型的如 MOS-ELM 算法[32]、ELM 算法[33]、ADL 算法[34]、AO-LEM 算法[35]等. 结构稳定的在线深度网络并不对网络架构进行直接调整, 而是通过集成学习思想将多个(层)网络结合起来, 通过调整网络权重来适应流数据中的概念漂移, 典型的如 HBP 算法[36]、SEOA 算法[37]等.

本文将梯度提升思想引入含概念漂移的流数据挖掘问题, 提出了一种基于自适应深度集成网络的概念漂移收敛方法, 通过集成多个浅层网络, 将后序基学习器的输出加到前序输出上对损失函数进行优化, 以提升学习模型的实时泛化性能. 与传统方法相比, 该方法通过对前序学习器输出进行不断纠错, 加速模型收敛, 并为解决含概念漂移的流数据挖掘问题提供了一个新思路.

2 基于自适应深度集成网络的概念漂移收敛方法

尽管深度学习在图像分类、目标检测、机器翻译等领域取得了较大成功，但在流数据场景下，深度学习算法难以通过验证集来对模型的架构进行选择，深层网络与浅层网络分别适用于不同场景，在线深度学习算法需要根据数据流的变化实时调整自身架构以达到收敛速度与表征能力间的有效平衡.

为解决上述问题，本文将梯度提升思想的纠错机制引入含概念漂移的流数据分类中，提出了一种基于自适应深度集成网络的概念漂移收敛方法. 该方法集成多个浅层网络作为基学习器，后序基学习器在前序基学习器输出的基础上进行纠错，虽然浅层网络表征能力较弱，难以拟合复杂的非线性映射，但是通过集成多个浅层网络进行多轮纠错操作，最终的学习模型在经过多轮更新后能够得到较好的实时泛化性能. 此外，由于浅层网络收敛速度快，模型能够较快地从概念漂移造成的精度下降中恢复过来，较好地解决了其他在线深度学习方法在处理含概念漂移的流数据问题时难以兼顾模型精度与恢复速度这一问题，从而能够适用于含概念漂移的流数据挖掘任务之中. 模型的基本架构如图 1 所示.

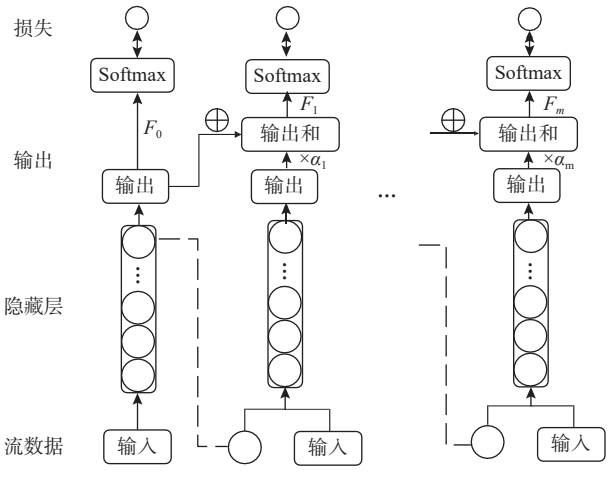


Fig. 1 Framework of concept drift convergence method based on adaptive deep ensemble networks

图 1 基于自适应深度集成网络的概念漂移收敛方法框架图

2.1 模型架构

基于自适应深度集成网络的概念漂移收敛方法集成 m 个浅层网络作为基学习器，第 i 个基学习器的输出为 $f_i(\boldsymbol{x},\theta_i), i=0,1,\cdots,m$，其中 θ_i 为第 i 个基学习器的对应参数，对于分类问题，这里的输出指 Softmax 操作之前的网络输出，每个基学习器对应一个权重 α_i，因此模型的输出为

$$F_m(\boldsymbol{x})=\sum_{i=0}^{m}\alpha_i f_i(\boldsymbol{x},\theta_i). \tag{1}$$

前序学习器的隐藏层输出与原始输入 $\boldsymbol{x}$ 合并在一起作为后序基学习器的输入. 由于后序基学习器的输入部分来自前序基学习器，因此每添加 1 个基学习器可视为增加了 1 层网络深度，即对输入数据进一步地处理与变换，提高了模型容量. 通过各学习器的实时性能来动态调整权重 α_i，以动态控制模型容量.

从图 1 可以看出，后序基学习器的任务相当于在前序基学习器输出的基础上进行纠错，假设

$$F_{m-1}(\boldsymbol{x})=\sum_{i=0}^{m-1}\alpha_i f_i(\boldsymbol{x},\theta_i) \tag{2}$$

为第 m−1 个基学习器的输出，那么第 m 个基学习器的学习目标即为在添加对应的输出以及权重之后的输出：

$$F_m(\boldsymbol{x})=\sum_{i=0}^{m}\alpha_i f_i(\boldsymbol{x},\theta_i), \tag{3}$$

能够进一步减小损失函数，即在添加上后序基学习器的输出之后，能够对前序输出纠错，以得到较好的实时泛化性能. 在该模型中，即使单个浅层网络难以达到较好的性能，但在添加上多个基学习器的输出，经过多轮纠错操作后的实时泛化性能可能较之前有明显提升. 此外，由于浅层网络的收敛速度较快，因此所提出的模型能够较快地从概念漂移造成的性能下降中恢复，有效兼顾了模型性能与收敛速度，可有效处理含概念漂移的流数据分类.

2.2 纠错操作

为使添加后序基学习器输出后损失函数的结果有所下降，梯度提升思想借鉴最优化理论中梯度下降算法，每个基学习器相应的损失函数利用前序输出的梯度信息来进行优化，以提升模型性能.

设训练样本为 $\{y_i,\boldsymbol{x}_i\}_{i=1}^{n}$，梯度提升算法希望求得参数集 $\{\alpha_j,\theta_j\}_{j=1}^{m}$，使得

$$\{\alpha_j,\theta_j\}_{j=1}^{m}=\arg\min_{\{\alpha'_j,\theta'_j\}_{j=1}^{m}}\sum_{i=1}^{N}L\Big(y_i,\sum_{j=0}^{m}\alpha'_j f_j(\boldsymbol{x}_i,\theta'_j)\Big), \tag{4}$$

其中 L 为损失函数. 为做到这一点，梯度提升算法采用梯度下降法近似逼近最优解，对于 $j=1,2,\cdots,m$，令

$$g_j(\boldsymbol{x}_i)=\left[\frac{\partial L(y_i,F(\boldsymbol{x}_i))}{\partial F(\boldsymbol{x}_i)}\right]_{F(\boldsymbol{x})=F_{m-1}(\boldsymbol{x})} \tag{5}$$

代表第 j 个基学习器上输出损失的梯度. 由于梯度下降只关心梯度的方向信息而不关心梯度的具体大小，因此这里可以设置一个缩放因子 β，按照式(6)训练第 j 个基学习器：

$$\theta_j=\arg\min_{\theta,\beta}\sum_{i=1}^{n}\left[-g_j(\boldsymbol{x}_i)-\beta f_j(\boldsymbol{x}_i,\theta)\right]^2. \tag{6}$$

在训练好第 j 个基学习器后，利用线搜索来求得梯度下降步长对应权重 α_j：

$$\alpha_j=\arg\min_{\alpha}\sum_{i=1}^{n}L(y_i,F_{j-1}(\boldsymbol{x}_i)+\alpha f_j(\boldsymbol{x}_i,\theta_j)). \tag{7}$$

在求得步长之后，利用式(8)更新模型：

$$F_j(\boldsymbol{x})=F_{j-1}(\boldsymbol{x})+\alpha_j f_j(\boldsymbol{x},\theta_j). \tag{8}$$

在经过 m 步梯度下降，即训练好 m 个基学习器之后,即可得到最终的模型.

然而，梯度提升思想不能直接应用于流数据挖掘中，这是由于流数据分类的高实时性要求模型能够 1 次只训练 1 个或一小批数据，且数据流分布动态变化，这使得后序基学习器输出的梯度信息包含较大误差，难以对损失函数在前序输出处的梯度信息进行准确学习. 为此，本文改进了梯度提升机思想，使其能够适用于含概念漂移的流数据分类.

具体地，在分类任务中，假设在时刻 t 基学习器 $f_0,f_1,\cdots,f_{j-1}$ 的参数均已更新完毕，f_j 的参数按照式(9)更新，即首先利用时刻 $t-1$ 的参数 θ_j^{t-1} 计算

$$\tilde{F}=\sum_{i=0}^{j-1}\alpha_i^t f_i(\boldsymbol{x},\theta_i^t)+\alpha_j^{t-1}f_j(\boldsymbol{x},\theta_j^{t-1}), \tag{9}$$

即先添加时刻 $t-1$ 的模型 f_j^{t-1} 的输出结果，并令

$$\tilde{y}_t=Softmax(\tilde{F}_j(\boldsymbol{x})) \tag{10}$$

为 f_j 的分类输出，利用

$$\theta_j^t=\theta_j^{t-1}-\eta\frac{\partial L(y_t,\tilde{y}_t)}{\partial\theta_j^{t-1}} \tag{11}$$

以及

$$\alpha_j^t=\alpha_j^{t-1}-\eta\frac{\partial L(y_t,\tilde{y}_t)}{\partial\alpha_j^{t-1}} \tag{12}$$

对 f_j 的参数以及权重进行更新，将更新后的 f_j 添加到模型中即得到

$$F_j(\boldsymbol{x})=\sum_{i=0}^{j}\alpha_i^t f_i(\boldsymbol{x},\theta_i^t), \tag{13}$$

在经过 m 轮更新后，模型最终的输出结果即为 $Softmax(F_m(\boldsymbol{x}))$.

该方法通过直接对损失函数进行优化来更新各基学习器的参数，使得更新后的基学习器损失减小. 相较于传统的梯度提升思想通过学习梯度信息来减小损失的方式，本文直接对损失函数优化，避免了流数据挖掘中基学习器难以准确学习梯度信息的问题.

2.3 算法总结

根据流数据到达的先后顺序将数据块赋予一个时间戳，流数据可以表示为

$$SD=\{(\boldsymbol{X}_t,Y_t)|t=1,2,\cdots,T\}.$$

按照流数据的这种表示形式，可以将本文所提出的基于自适应深度集成网络的概念漂移收敛方法总结如算法 1 所示.

算法 1. 基于自适应深度集成网络的概念漂移收敛算法.

输入：初始化各基学习器，设每个学习器输出的 logits 为 $f_i(\boldsymbol{x},\theta_i),i=0,1,\cdots,m$，并且将权重初始化为 $\alpha_i=1/m,i=1,2,\cdots,m$；

输出：模型输出结果 $Softmax(F_{\mathrm{m}}(\boldsymbol{x}))$.

① for $t=1,2,\cdots,T$

② 预测 $\tilde{Y}_t=Softmax(f_0(\boldsymbol{X}_t,\theta_0^{t-1})+\sum_{j=1}^{m}\alpha_j^{t-1}f_j(\boldsymbol{X}_t,\theta_j^{t-1}))$；

③ 令 $\tilde{Y}_0^t=Softmax(f_0(\boldsymbol{X}_t,\theta_0^{t-1}))$，并利用 $\theta_0^t=\theta_0^{t-1}-\eta\frac{\partial L(Y_t,\tilde{Y}_0^t)}{\partial\theta_0^{t-1}}$ 来更新 θ_0；

④ for $j=1,2,\cdots,m$

⑤ $\tilde{Y}_j^t=Softmax(\sum_{i=0}^{j-1}\alpha_i^t f_i(\boldsymbol{x},\theta_i^t)+\alpha_j^{t-1}f_j(\boldsymbol{x},\theta_j^{t-1}))$；

⑥ 更新参数 $\theta_j^t=\theta_j^{t-1}-\eta\frac{\partial L(Y_t,\tilde{Y}_j^t)}{\partial\theta_j^{t-1}}$；

⑦ 更新参数 $\alpha_j^t=\alpha_j^{t-1}-\eta\frac{\partial L(Y_t,\tilde{Y}_j^t)}{\partial\alpha_j^{t-1}}$；

⑧ end for

⑨ end for

3 实验结果与分析

为测试本文所提出的 CD_ADEN 方法的性能，本文在多个真实数据集以及合成数据集上对模型的实时精度、累积精度及概念漂移发生后的恢复性等评价指标进行了测试. 所采用的深度学习框架为 Tensorflow，实验所用计算机的配置为 Intel® Core™ i5-8300H 2.30 GHz CPU, 16 GB DDR4L RAM 以及 NVIDIA GeForce GTX1050 Ti 4GB GPU.

3.1 数据集及对比方法

本文所用的数据集既有真实数据集，又有模拟概念漂移的合成数据集，包含了突变型概念漂移与渐变型概念漂移，数据集[37]的具体信息见表 1.

Table 1 Datasets Used in Experiment

表 1 实验采用的数据集

数据集	样本实例数	属性维数	样本类别数	漂移类型	漂移位点位置
Sea	10^5	3	2	渐变型	25×10^3，50×10^3，75×10^3
Hyperplane	10^5	10	2	增量型	-
RBFBlips	10^5	20	4	突变型	25×10^3，50×10^3，75×10^3
LED_abrupt	10^5	24	10	突变型	50×10^3
LED_gradual	10^5	24	10	渐变型	25×10^3，50×10^3，75×10^3
Tree	10^5	30	10	突变型	25×10^3，50×10^3，75×10^3
Electricity	4.53×10^4	6	2	未知	-
Kddcup99	4.94×10^6	41	23	未知	-
Covertype	5.81×10^5	54	7	未知	-
Weather	9.51×10^4	9	3	未知	-

注：“-”表示不确定漂移位点位置.

在本文中将所提出的方法 CD_ADEN 与 4 种在线深度学习方法 HBP[36], Resnet[38], Highway[39], DNN 进行了对比，实验中数据块大小为 100，隐藏节点为 100，使用 ReLU 激活函数，设置固定学习率为 0.001.

3.2 评价指标

本文从 3 个指标对模型的泛化性能及概念漂移发生后模型的收敛性能进行测试与分析.

1）平均实时精度. 平均实时精度即实时精度在每个时间节点上的平均值，定义为

$$Avgracc=\frac{1}{T}\sum_{t=1}^{T}acc_t, \tag{14}$$

其中acc_t为模型在时间节点 t 的实时精度. 平均实时精度越高，反映模型的实时性能越好. 由于本文使用的数据集多为类别不平衡的数据集，因此本文中的实时精度acc_t使用的是 Balanced Accuracy Score[40]，acc_t被定义为各类别的平均值：

$$acc_t=\frac{1}{k}\sum_{i=1}^{k}\frac{c_{ii}}{c_{*i}}, \tag{15}$$

其中 k 为类别数目，c_{ij}指混淆矩阵中索引为(i,j)的元素，$c_{*i}=\sum_{j=1}^{k}c_{ji}$为混淆矩阵第 i 列元素之和.

2）最终累积精度. 最终累积精度即最终模型预测正确的样本数与总样本数的比值，即

$$Finacc=\frac{1}{T\times n}\sum_{t=1}^{T}n_t, \tag{16}$$

其中 n 为每个数据块中的数据数量，n_t为模型在时间节点 t 对应的数据块中分类正确的样本数目. 最终累积精度越高，代表模型的整体分类性能越好.

3）漂移恢复率. 一个好的在线学习算法不仅需要有较高的准确率，并且需要能够较快地从概念漂移造成的影响中恢复，因此漂移恢复率 Drr 指标定义为

$$Drr=step\times avg, \tag{17}$$

其中$step$代表模型从概念漂移造成的影响中恢复所需的时间步长，avg代表恢复过程中模型的平均错误率. 漂移恢复率取值越小，表示模型在漂移发生后，越能够快速恢复到漂移之前的性能.

3.3 实验结果及分析

为验证本文所提方法的合理性，本节从实时精度、累积精度以及漂移恢复率 3 个指标进行了实验分析.

3.3.1 实时精度的结果与分析

模型的超参数往往对模型的性能有着至关重要的影响，本节在不同数据集上分析基学习器数量下模型的性能表现. 表 2 展示了不同基学习器数量下模型的平均实时精度情况，本文分别测试了包含 4 个、8 个、12 个基学习器时模型的平均实时精度. 从表 2 可以看出，在多数数据集上，基学习器数量 n>8 时，性能差距不明显. 因此，后续与其他方法对比时本文所提出的模型均包含 8 个基学习器.

Table 2 Average Real-Time Accuracy with Different Numbers of Base Learners

表 2 不同基学习器数量下的平均实时精度

数据集	平均实时精度（排名）		
	n=4	n=8	n=12
Sea	0.822 1（3）	**0.824 3（1）**	0.822 6（2）
Hyperplane	0.919 5（3）	0.920 0（2）	**0.921 4（1）**
RBFBlips	0.967 4（3）	0.968 0（2）	**0.970 5（1）**
LED_abrupt	0.612 3（2）	0.612 0（3）	**0.613 6（1）**
LED_gradual	**0.624 9（1）**	0.622 7（3）	0.623 2（2）
Tree	0.385 8（3）	0.389 5（2）	**0.391 6（1）**
Electricity	0.646 8（3）	0.648 4（2）	**0.648 5（1）**
Kddcup99	**0.842 4（1）**	0.837 2（2）	0.819 9（3）
Covertype	0.703 9（2）	**0.705 6（1）**	0.703 8（3）
Weather	0.812 4（3）	**0.812 5（1）**	0.812 5（2）

注：n 表示基学习器数量；黑体数值表示最优值.

图 2 展示了不同方法的实时精度变化趋势. 从图 2 中可以看出，不同方法实时精度波动趋势基本一致，但 CD_ADEN 方法波动较小，因此较好地处理了流数据中包含的不同类型概念漂移.

表 3 展示了不同方法的平均实时精度以及相应排序情况. 可以看出，在多数情况下 CD_ADEN 方法

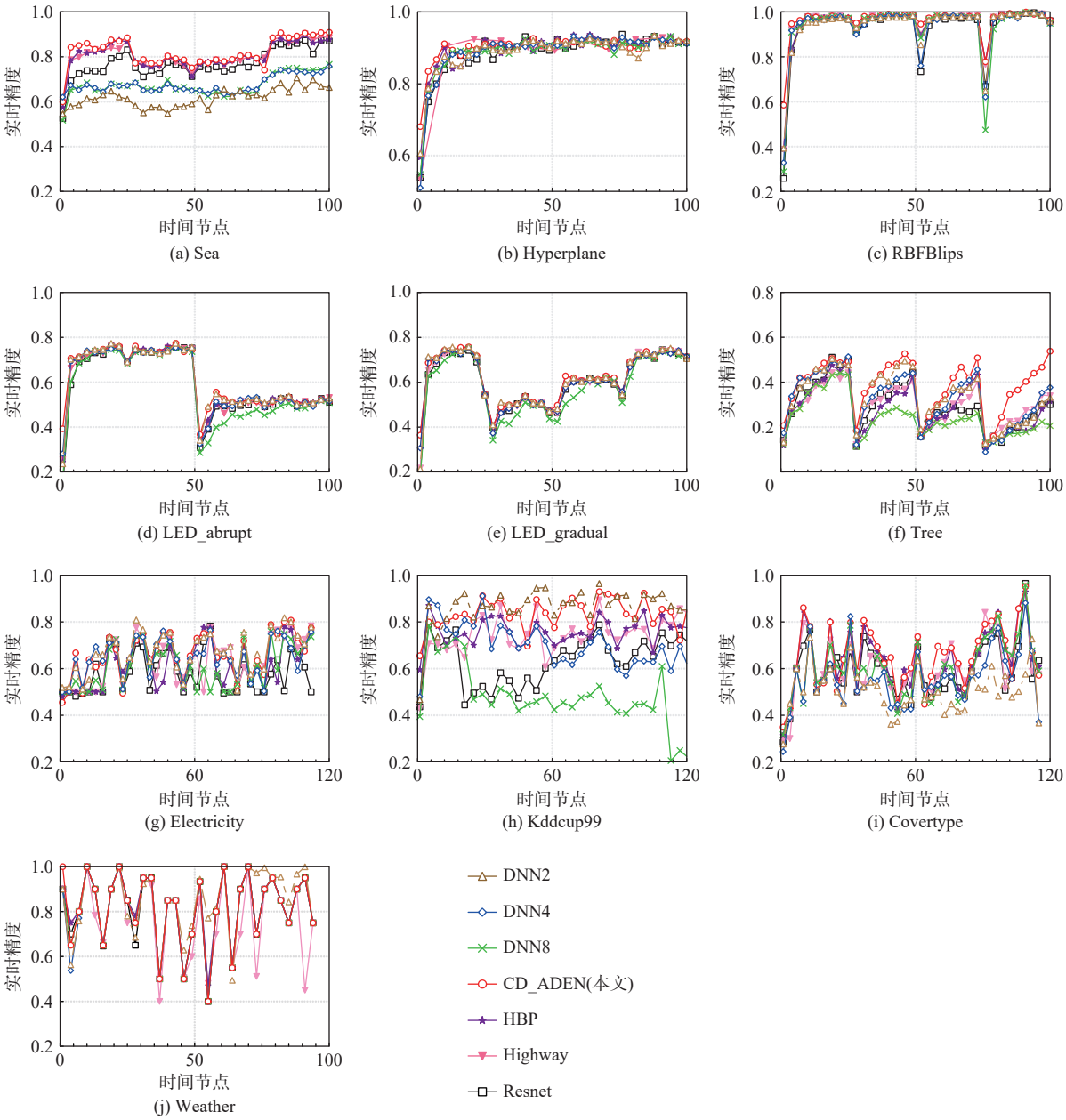


Fig. 2 Comparison of real-time accuracy of different methods

图 2 不同方法的实时精度比较

的平均实时精度最好. 然而在 LED_abrupt, Electricity, Kddcup99 和 Weather 上，CD_ADEN 方法并未取得最优结果，这可能是由于数据集的分布不同，较少基学习器模型不能很好地表示数据分布，而较多基学习器则会降低收敛速度，并且导致较低的平均实时精度.

本文通过非参数测试方法 Friedman 检验[41]进行统计测试. 对于给定的 k 个算法和 n 个数据集，r_i^j 是第 j 个算法在第 i 个数据集上的排名，第 j 个算法的平均序值 $R_j=\frac{1}{n}\sum_i r_i^j$. 零假设 H_0 为如果所有的方法性能相同，那么它们的平均序值是相等的. 在零假设下，当 k 和 n 足够大时，F_{F} 服从自由度为 $k-1$ 和 $(k-1)(n-1)$ 的 F 分布：

$$F_{\mathrm{F}}=\frac{(n-1)\chi_{\mathrm{F}}^2}{n(k-1)-\chi_{\mathrm{F}}^2}, \tag{18}$$

其中

$$\chi_{\mathrm{F}}^2=\frac{12n}{k(k+1)}\left[\sum_j R_j^2-\frac{k(k+1)^2}{4}\right]. \tag{19}$$

若零假设 H_0 被拒绝，计算出的统计量大于 F_{F} 的临界值，表明学习方法在性能上有显著差异，对上述方法的平均实时精度进行测试，得到统计值 F_{F}=9.75.

Table 3 Average Real-Time Accuracy Comparison of Different Methods

表 3 不同方法的平均实时精度比较

数据集	平均实时精度（排名）						
	DNN2	DNN4	DNN8	Resnet	Highway	HBP	CD_ADEN（本文）
Sea	0.621 7（7）	0.678 2（6）	0.678 2（5）	0.769 8（4）	0.806 4（2）	0.805 6（3）	**0.824 3（1）**
Hyperplane	0.891 3（5）	0.893 0（4）	0.890 5（6）	0.889 2（7）	0.905 3（2）	0.896 9（3）	**0.920 0（1）**
RBFBlips	0.946 0（4）	0.944 8（6）	0.945 1（5）	0.940 6（7）	0.958 1（2）	0.951 7（3）	**0.968 0（1）**
LED_abrupt	**0.612 2（1）**	0.603 0（4）	0.574 1（7）	0.591 2（6）	0.601 6（5）	0.606 8（3）	0.612 0（2）
LED_gradual	0.620 6（2）	0.613 4（4）	0.587 3（7）	0.606 7（6）	0.611 6（5）	0.619 1（3）	**0.622 7（1）**
Tree	0.330 1（2）	0.326 8（3）	0.237 8（7）	0.287 7（5）	0.295 2（4）	0.279 9（6）	**0.389 5（1）**
Electricity	**0.665 7（1）**	0.646 5（3）	0.603 1（6）	0.582 1（7）	0.633 7（4）	0.611 5（5）	0.648 4（2）
Kddcup99	**0.879 6（1）**	0.718 6（6）	0.476 3（7）	0.653 4（5）	0.753 7（4）	0.767 0（3）	0.837 2（2）
Covertype	0.525 1（7）	0.573 9（6）	0.624 3（4）	0.618 3（5）	0.635 4（3）	0.646 5（2）	**0.705 6（1）**
Weather	**0.847 8（1）**	0.805 0（5）	0.805 7（4）	0.803 4（6）	0.781 3（7）	0.813 9（2）	0.812 5（3）
平均序值	3.1	4.7	5.8	5.8	3.8	3.3	**1.5**

注：黑体数值表示最优值.

由于其在显著水平α=0.05 处的临界值为 2.272, 因此拒绝了算法之间性能不可区分的零假设.

此外, 所有方法的临界差(CD)都是通过 Bonferroni-Dunn 检验计算的, 用来显示 CD_ADEN 方法和对比方法之间的相对性能, 如果 2 个算法的平均序值之差超过了临界差 CD, 则 2 个分类器的性能有显著差异:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6n}}, \tag{20}$$

其中q_α是显著级α的临界值. 可得在α=0.05 时 CD=2.195. 统计分析的结果如图 3 所示. 其中, 平均序值在一个临近值域内的方法用黑线连接. 结果表明, CD_ADEN 方法的平均实时精度显著优于 DNN8, Resnet, DNN4.

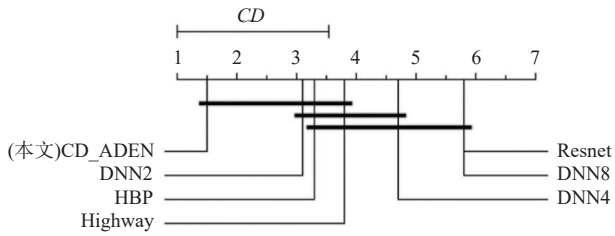


Fig. 3 Bonferroni-Dunn test result for average real-time accuracy

图 3 平均实时精度的 Bonferroni-Dunn 检验结果

3.3.2 累积精度的结果与分析

表 4 展示了不同基学习器数量 n 下模型的最终累积精度情况. 在大多数据集上, 基学习器数量越大, 模型的最终累积精度表现越好.

图 4 展示了不同方法的累积精度比较. 可以看出所有方法的累积精度趋势相同, 漂移发生后, CD_ADEN 的累积精度下降低于其他方法, 这是由于后序基学习器在前序输出基础上不断纠错, 提升了模型的泛化性能.

Table 4 Final Cumulative Accuracy with Different Numbers of Base Learners

表 4 不同基学习器数量下的最终累积精度

数据集	最终累积精度（排名）		
	n=4	n=8	n=12
Sea	0.845 5（3）	**0.847 2（1）**	0.846 3（2）
Hyperplane	0.890 9（3）	**0.897 9（1）**	0.893 6（2）
RBFBlips	0.965 1（3）	0.967 3（2）	**0.968 1（1）**
LED_abrupt	**0.613 3（1）**	0.612 8（3）	0.613 1（2）
LED_gradual	0.623 9（2）	0.622 6（3）	**0.624 4（1）**
Tree	**0.595 2（1）**	0.593 7（2）	0.576 9（3）
Electricity	**0.687 4（1）**	0.678 1（2）	0.661 1（3）
Kddcup99	0.983 0（2）	**0.984 4（1）**	0.981 9（3）
Covertype	**0.812 3（1）**	**0.812 3（1）**	0.802 9（3）
Weather	**0.883 8（1）**	**0.883 8（1）**	**0.883 8（1）**

注：n 表示基学习器数量；黑体数值表示最优值.

表 5 展示了不同方法的最终累积精度结果以及相应的排序. 通过平均序值的比较可以看出, CD_ADEN 的性能最好, HBP 和 DNN2 次之, DNN8 表现最差. CD_ADEN 在除了 Hyperplane 的合成数据集上都取得最高的最终累积精度, 在真实数据集 Electricity, Weather 上表现稍差于 DNN2 方法. CD_ADEN 通过将梯度提升算法的纠错机制引入流数据挖掘任务中处理概念漂移, 使模型具有较好的泛化性能. 然而, CD_ADEN 并不是均能保持最优精度, 这可能是由于

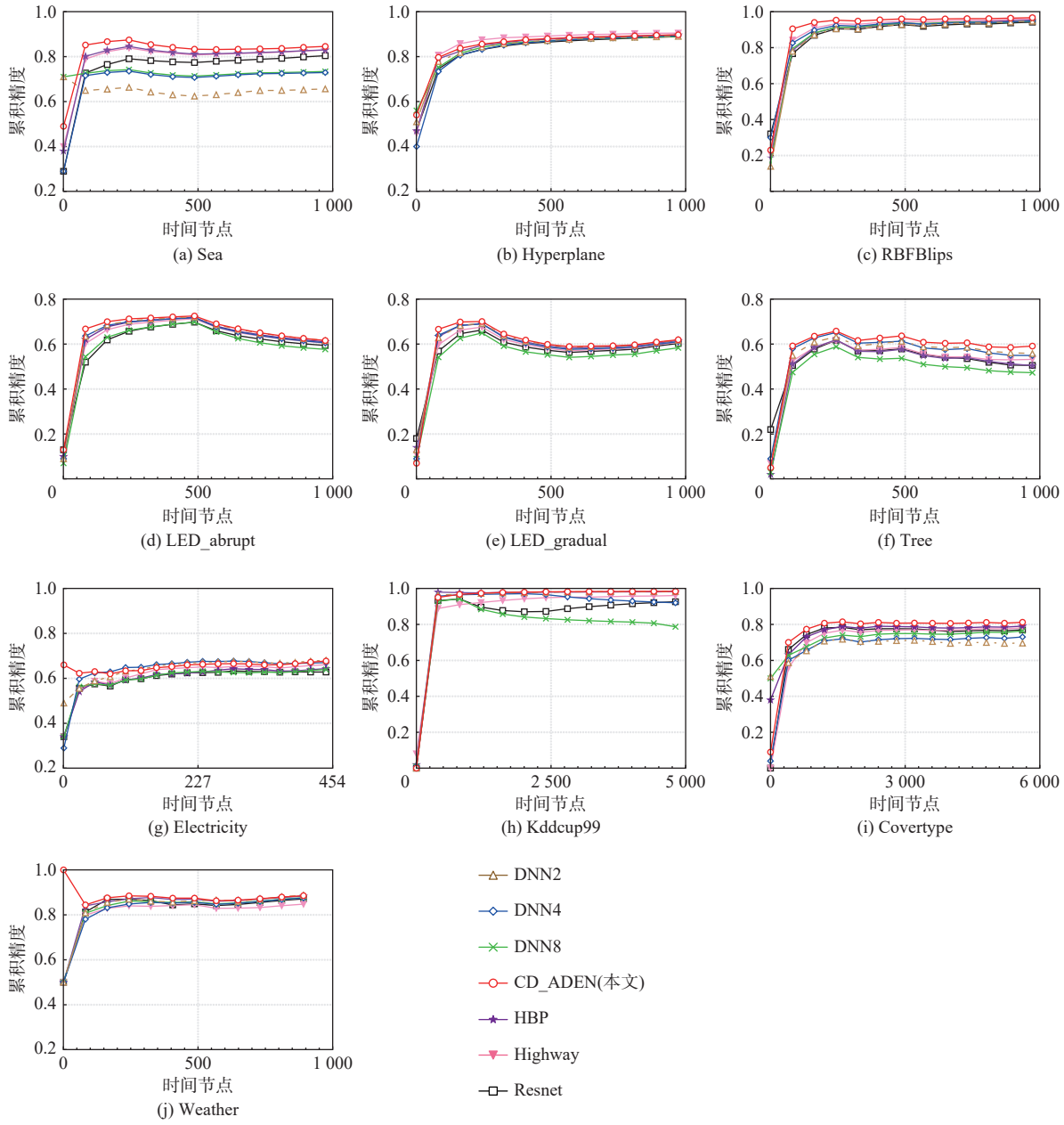


Fig. 4 Comparison of cumulative accuracy on different methods

图 4 不同方法的累积精度比较

基学习器数量未选择至合适数目, 较少或较多的基学习器数量均会导致在部分数据集上取得较低的最终累积精度.

进一步分析计算出检验值 F_F=10.874 4, 在显著水平 α=0.05 处的临界值为 2.109, 因此拒绝了方法之间性能不可区分的零假设. Bonferroni-Dunn 检验结果如图 5 所示, 结果表明 CD_ADEN 方法的最终累积精度显著优于 DNN4, DNN8, Resnet.

3.3.3 漂移恢复率的结果与分析

在线学习模型不仅要考虑其准确率, 同时也应该考虑概念漂移发生之后模型的恢复性能. 本文在突变型概念漂移数据集上对模型的恢复性能进行了测试. 当模型的实时精度恢复到概念漂移发生之前的 δ 倍时即判断为恢复, 根据数据集的不同, δ 相应被设置成不同的值.

表 6 展示了不同基学习器数量 n 下模型的漂移恢复性能. 由表 6 可以看出, 不同于多层 DNN, 当模型的基学习器越多, 就对应着更复杂、容量更大的模型, 模型的恢复速度反而更快. 这可能是因为越多的基学习器对应越多的纠错次数, 可以使模型更快地从概念漂移导致的高错误率中恢复. 这也一定程度上印证了本文提出的纠错机制是适用于含概念漂移

Table 5 Final Cumulative Accuracy Comparison of Different Methods

表 5 不同方法的最终累积精度比较

数据集	最终累积精度（排名）						
	DNN2	DNN4	DNN8	Resnet	Highway	HBP	CD_ADEN（本文）
Sea	0.657 9（7）	0.730 3（6）	0.734 9（5）	0.806 2（4）	0.830 7（3）	0.830 9（2）	**0.847 2（1）**
Hyperplane	0.891 2（5）	0.893 3（4）	0.890 8（6）	0.888 7（7）	**0.905 3（1）**	0.897 0（3）	0.897 9（2）
RBFBlips	0.947 9（5）	0.948 9（4）	0.947 8（6）	0.943 3（7）	0.960 0（2）	0.954 2（3）	**0.967 3（1）**
LED_abrupt	0.612 5（2）	0.603 3（4）	0.574 7（7）	0.591 5（6）	0.601 6（5）	0.606 9（3）	**0.612 8（1）**
LED_gradual	0.620 8（2）	0.613 7（4）	0.587 9（7）	0.606 7（6）	0.611 7（5）	0.618 6（3）	**0.622 6（1）**
Tree	0.558 9（2）	0.550 5（3）	0.472 5（7）	0.506 9（5）	0.534 0（4）	0.504 2（6）	**0.593 7（1）**
Electricity	**0.682 1（1）**	0.671 5（3）	0.641 1（6）	0.630 3（7）	0.662 1（4）	0.650 4（5）	0.678 1（2）
Kddcup99	0.983 2（2）	0.919 5（6）	0.781 2（7）	0.927 6（5）	0.961 4（4）	0.982 3（3）	**0.984 4（1）**
Covertype	0.698 4（7）	0.733 6（6）	0.767 7（5）	0.773 0（4）	0.782 4（3）	0.790 3（2）	**0.812 3（1）**
Weather	**0.887 2（1）**	0.874 3（5）	0.875 4（4）	0.870 8（6）	0.836 2（7）	0.882 4（3）	0.883 8（2）
平均序值	3.4	4.5	6.0	5.7	3.8	3.3	**1.3**

注：黑体数值表示最优值.

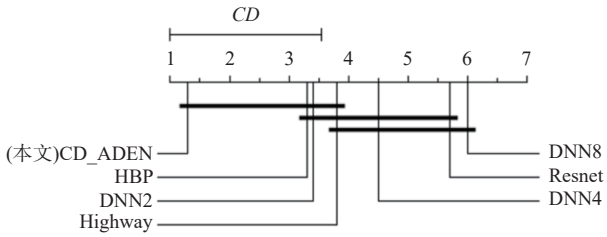


Fig. 5 Bonferroni-Dunn test result for final cumulative accuracy

图 5 最终累积精度的 Bonferroni-Dunn 检验结果

的流数据挖掘问题的解决之中的.

表 7 展示了 CD_ADEN 与其他方法的对比. 可以看出, CD_ADEN 有较好的恢复速度, 平均排名为各种方法中最高, 能够较好地应用于含概念漂移的流数据挖掘任务.

进一步分析上述结果, 比较 CD_ADEN 和对比方法在漂移恢复方面的性能. 计算出 F_F=6.501 6, 在显著水平α=0.05 处的临界值为 2.109, 因此拒绝了方法之

Table 6 Drift Recovery Rate Under Different Numbers of Base Learners

表 6 不同基学习器数量下的漂移恢复率

数据集	漂移恢复率（第 1 个位点/第 2 个位点/第 3 个位点）		
	n=4	n=8	n=12
Tree（δ = 0.7）	**33.37**/69.35/112.14	40.39/84.51/**66.52**	40.81/**42.49**/80.80
RBFBlips（δ = 0.9）	1.60/**1.91**/1.35	**1.59**/2.16/1.36	1.72/2.09/**1.10**
LED_abrupt（δ = 0.7）	40.06	31.98	**28.77**

注：LED_abrupt 数据集包含 1 个位点，漂移恢复率只有 1 个值；n 表示基学习器数量. 黑体数值表示不同位点的最优值.

Table 7 Drift Recovery Rate of Different Methods

表 7 不同方法的漂移恢复率

方法	漂移恢复率（第 1 个位点/第 2 个位点/第 3 个位点）			平均序值
	Tree（δ = 0.7）	RBFBlips（δ = 0.9）	LED_abrupt（δ = 0.7）	
DNN2	45.98/88.41/193.43	4.32/4.39/3.18	**27.64**	3.71
DNN4	75.89/90.89/166.07	4.76/6.86/4.95	46.48	5.0
DNN8	171.02/**53.77**/158.23	3.55/3.31/6.45	124.98	4.43
Resnet	98.63/176.26/126.73	5.04/7.47/2.81	46.94	5.29
Highway	59.56/97.21/147.49	3.62/2.89/1.86	45.97	3.29
HBP	113.92/104.81/194.90	5.43/3.43/1.43	45.38	5.0
CD_ADEN（本文）	**40.39**/80.51/**66.52**	**1.59**/**2.16**/**1.36**	31.98	**1.29**

注：同表 6，LED_abrupt 数据集的漂移恢复率只有 1 个值. 黑体数值表示不同位点的最优值.

间性能不可区分的零假设. Bonferroni-Dunn 检验结果如图 6 所示, 结果表明, CD_ADEN 方法的漂移恢复性能显著优于 DNN4, DNN8, Resnet.

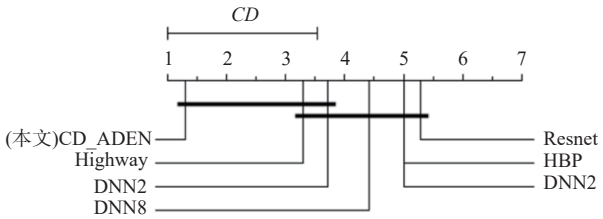


Fig. 6 Bonferroni-Dunn test result for drift recovery rate

图 6 漂移恢复率的 Bonferroni-Dunn 检验结果

4 总结与展望

本文针对流数据挖掘中概念漂移问题带来的挑战, 将梯度提升的纠错思想引入概念漂移问题的解决之中, 提出了基于自适应深度集成网络的概念漂移收敛方法 CD_ADEN. CD_ADEN 方法通过集成多个浅层网络构成集成学习模型, 后序基学习器在前序输出的基础上对其进行纠错以提升模型的实时泛化性能, 在一定程度上缓解了传统在线深度网络难以兼顾模型精度与恢复性能的问题, 从而更好地应用于含概念漂移的流数据挖掘任务当中.

作者贡献声明: 郭虎升提出思路, 设计方法, 负责初稿写作及论文修改; 孙妮负责论文撰写、数据测试及论文修改; 王嘉豪负责代码实现、数据测试及论文撰写; 王文剑负责写作指导、论文修改审定.

参 考 文 献

[1] Lu Jie, Liu Anjin, Dong Fan, et al. Learning under concept drift: A review[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 31(12): 2346−2363

[2] Wen Yimin, Tang Shiqi, Feng Chao, et al. Recurrent concept drift data stream classification based on online transfer learning[J]. Journal of Computer Research and Development, 2016, 53(8): 1781−1791(in Chinese)
(文益民, 唐诗淇, 冯超, 等. 基于在线迁移学习的重现概念漂移数据流分类 [J]. 计算机研究与发展, 2016, 53(8): 1781−1791)

[3] Zhai Tingting, Gao Yang, Zhu Junwu. A review of online learning 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)

[4] Lu Yang, Cheung Y M, Tang Yangyang. 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

[5] Sun Yu, Tang Ke, Zhu Zexuan, et al. Concept drift adaptation by exploiting historical knowledge[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(10): 4822−4832

[6] 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

[7] Brzeziński D, Stefanowski J. Accuracy updated ensemble for data streams with concept drift [C] // Proc of the 6th Int Conf on Hybrid Artificial Intelligence Systems. Berlin: Springer, 2011: 155−163

[8] Mohamad S, Sayed-Mouchaweh M, Bouchachia A. Active learning for classifying data streams with unknown number of classes[J]. Neural Networks, 2018, 98: 1−15

[9] Bu Li, Alippi C, Zhao Dongbin. A pdf-free change detection test based on density difference estimation[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(2): 324−334

[10] Ross G J, Adams N M, Tasoulis D K, et al. Exponentially weighted moving average charts for detecting concept drift[J]. Pattern Recognition Letters, 2012, 33(2): 191−198

[11] Cai Huan, Lu Kezhong, Wu Qirong, et al. Adaptive classification algorithm for concept drift data streams[J]. Journal of Computer Research and Development, 2022, 59(3): 633−646(in Chinese)
(蔡桓, 陆克中, 伍启荣, 等. 面向概念漂移数据流的自适应分类算法 [J]. 计算机研究与发展, 2022, 59(3): 633−646)

[12] Bifet A, Holmes G, Pfahringer B, et al. New ensemble methods for evolving data streams [C] // Proc of the 15th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2009: 139−148

[13] Brzezinski D, Stefanowski J. Reacting to different types of concept drift: The accuracy updated ensemble algorithm[J]. IEEE Transactions on Neural Networks and Learning Systems, 2014, 25(1): 81−94

[14] Junior J R. Graph embedded rules for explainable predictions in data streams[J]. Neural Networks, 2020, 129: 174−192

[15] Guo Husheng, Zhang Aijuan, Wang Wenjian. Concept drift detection method based on online performance testing[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)

[16] Shen Yanning, Chen Tianyi, Giannakis G B. Random feature-based online multi-kernel learning in environments with unknown dynamics[J]. Journal of Machine Learning Research, 2019, 20(1): 773−808

[17] Du Hangyuan, Wang Wenjian, Bai Liang. An evolutionary 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)

[18] Sun Hao, Zheng Xiangtao, Lu Xiaoqing. A supervised segmentation network for hyperspectral image classification [J]. IEEE Transactions on Image Processing, 2021, 30: 2810−2825

[19] Tian Yicong, Dehghan Y, Shah M. On detection, data association and segmentation for multi-target tracking[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(9): 2146−2160

[20] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C] // Proc of the 31st Annual Conf on Neural Information Processing

Systems. Cambridge, MA: MIT, 2017: 5998−6008
[21] Rubio J, Angelov P, Pacheco J. Uniformly stable backpropagation algorithm to train a feedforward neural network[J]. IEEE Transactions on Neural Networks, 2011, 22(3): 356−366
[22] Chen Wenzheng, Mirdehghan P, Fidler S, et al. Auto-tuning structured light by optical stochastic gradient descent [C] // Proc of the 2020 IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 5969−5979
[23] Lu Ning, Lu Jie, Zhang Guangquan, et al. A concept drift-tolerant case-base editing technique[J]. Artificial Intelligence, 2016, 230: 108−133
[24] Zhu Qun, Hu Xuegang, Zhang Yuhong, et al. A double-window based classification algorithm for concept drifting data streams [C] // Proc of the 2010 IEEE Int Conf on Granular Computing. Piscataway, NJ: IEEE, 2010: 639−644
[25] Yang Ying, Wu Xindong, Zhu Xingquan. Combining proactive and reactive predictions for data streams [C] // Proc of the 11th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2015: 710−715
[26] Guo Husheng, Li Hai, Ren Qiaoyan, et al. Concept drift type identification based on multi-sliding windows[J]. Information Sciences, 2022, 585: 1−25
[27] Guo Husheng, Ren Qiaoyan, Wang Wenjian. Category detection of concept drift based on time series 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)
[28] 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
[29] Kolter J Z, Maloof M A. Dynamic weighted majority: An ensemble method for drifting concepts[J]. Journal of Machine Learning Research, 2007, 8: 2755−2790
[30] Zhao Peng, Zhou Zhihua. Data learning of distribution changes based on 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)
[31] Elwell R, Polikar R. Incremental learning of concept drift in non-stationary environments[J]. IEEE Transactions on Neural Networks, 2011, 22(10): 1517−1531
[32] Mirza B, Lin Zhipin. Meta-cognitive online sequential extreme learning machine for imbalanced and concept-drifting data classification[J]. Neural Networks, 2016, 80: 79−94
[33] Hammami Z, Mouelhi W, Said L B. Online self-adaptive framework for tailoring a neural-agent learning model addressing dynamic real-time scheduling problems[J]. Journal of Manufacturing Systems, 2017, 45: 97−108
[34] Ashfahani A, Pratama M. Autonomous deep learning: Continual learning approach for dynamic environments [C] // Proc of the 2019 SIAM Int Conf on Data Mining. Philadelphia, PA: SIAM, 2019: 666−674
[35] Yu Hualong, Webb G I. Adaptive online extreme learning machine by regulating forgetting factor by concept drift map[J]. Neurocomputing, 2019, 343: 141−153
[36] 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. San Francisco, CA: Morgan Kaufmann, 2018: 2660−2666
[37] 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
[38] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition [C] // Proc of the 2016 IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 770−778
[39] Srivastava R K, Greff K, Schmidhuber J. Training very deep networks [C] // Proc of the 2015 on Neural Information Processing Systems. Cambridge, MA: MIT, 2015: 2377−2385
[40] Brodersen K H, Ong C S, Stephan K E, et al. The balanced accuracy and its posterior distribution [C] // Proc of the 20th Int Conf on Pattern Recognition. Piscataway, NJ: IEEE, 2010: 3121−3124
[41] Demsar J. Statistical comparisons of classifiers over multiple datasets[J]. Journal of Machine Learning Research, 2006, 7(1): 1−3



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、机器学习及计算智能.



Sun Ni, born in 1998. Master. Her main research interests include stream data mining and online machine learning.
孙　妮, 1998 年生. 硕士. 主要研究方向为流数据挖掘、在线机器学习.



Wang Jiahao, born in 2000. Bachelor. His main research interest includes data mining.
王嘉豪, 2000 年生. 学士. 主要研究方向为数据挖掘.



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、数据挖掘及计算智能.