Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

流式数据漂移应对方法

硕士研究生 郭汝赞

2026年 05月 06日

问题回溯

- 总结反思
  - 语调平淡、语言信息量少
  - TIPO讲解少，没有突出创新点
- 相关内容
  - 2025.07.27 马西洋：《数据集不平衡评估方法》

内容提要

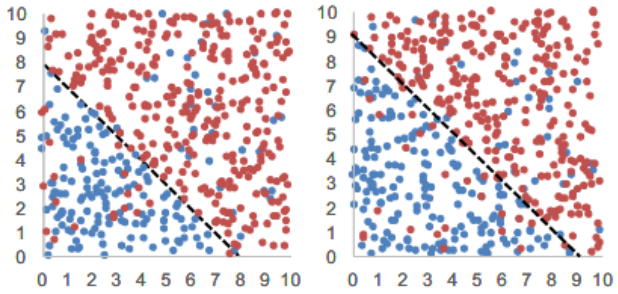
- 预期收获
- 内容引入
- 内涵解析与研究目标
- 研究背景与研究意义
- 研究历史与现状
- 知识基础
- 算法原理
  - PUDD
  - CALIPER
- 特点总结与工作展望
- 参考文献

预期收获

- 预期收获
  - 1. 了解基于数据漂移的基本概念和问题框架
  - 2. 理解数据漂移检测与重训练的必要性
  - 3. 了解数据漂移检测与重训练前沿方法

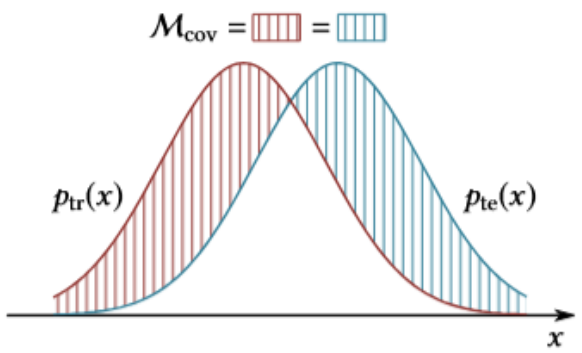
内涵解析与研究目标

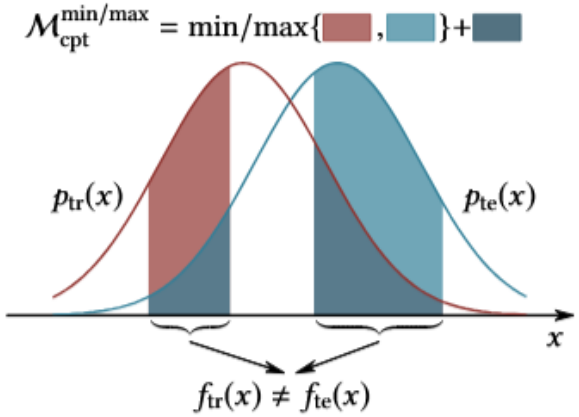
- 研究目标
  - 让机器学习模型在数据持续变化的环境中仍然保持可靠
    - 在模型性能明显下降前识别漂移
    - 在漂移后判断是否需要更新模型、何时更新模型



(a) Data at time t (b) Data at time $t+1$

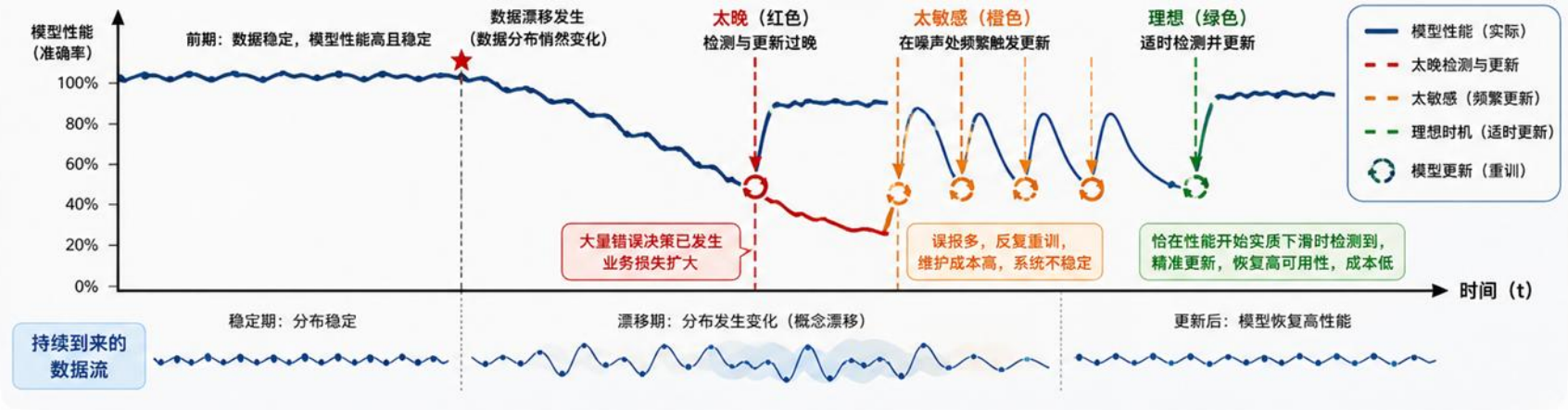
- 内涵解析
  - 数据漂移：数据流随时间变化后，模型原先学到的规律不再完全适用
  - 漂移检测：在连续到来的数据中比较过去和当前数据是否仍属于同一种规律





研究背景与研究意义

- 研究背景
  - 当前很多系统都在处理持续到来的数据，而不是处理静态数据集
  - 数据发生漂移时，模型可能表面仍在运行，但实际判断已经逐渐失准
    - 检测太晚会让错误决策持续发生
    - 检测太敏感又会造成频繁误报和反复重训
- 研究意义
  - 根据数据变化情况触发模型更新，提高真实环境中的长期可用性
  - 帮助系统及时发现模型失效风险、降低维护成本



研究历史

2007

Bifet提出了ADWIN，与需要预设窗口大小的算法不同，ADWIN核心是使用一个动态调整大小的数据窗口检测数据流中的漂移。它通过滑动窗口比较历史数据和当前数据，从而判断数据分布是否发生变化

2020

Christoph等人提出了KSWIN，它使用一个固定大小的滑动窗口，并持续比较窗口内两部分数据的分布，通过KS检验来判断数据分布是否发生了显著变化，但固定窗口对不同漂移速度不够灵活

2022

Yuan等人梳理了深度学习模型在漂移环境下的适应问题，他们梳理了突变漂移、渐进漂移、增量漂移和循环漂移，并把深度模型适应方法分为参数更新和结构更新两类，说明单纯检测漂移已经不够，还要解决深度模型如何稳定更新的问题

2023

Zhang等人提出了OneNet，认为不同模型对数据有不同偏好，而这种偏好会随着漂移变化，因此提出在线集成网络，动态组合多个预测模型，用来缓解时间序列预测中分布持续变化带来的性能下降，缺点是系统复杂度更高；对漂移解释性有限

2024

Wan等人出 MCD-DD，把窗口样本集合编码为概念嵌入，再利用最大概念差异和对比学习来检测高维数据流中的概念漂移，减少对标签、错误率或简单统计假设的依赖，更接近图像、复杂传感器和真实线上系统的需求，但训练和参数设置较复杂

2025

Lu等人提出了PUDD，首次理论上证明预测不确定性比错误率更敏感，它不依赖错误率下降，因此能实现早期检测，并且已在图像数据流上验证有效，不过概率校准质量会影响检测效果

2026

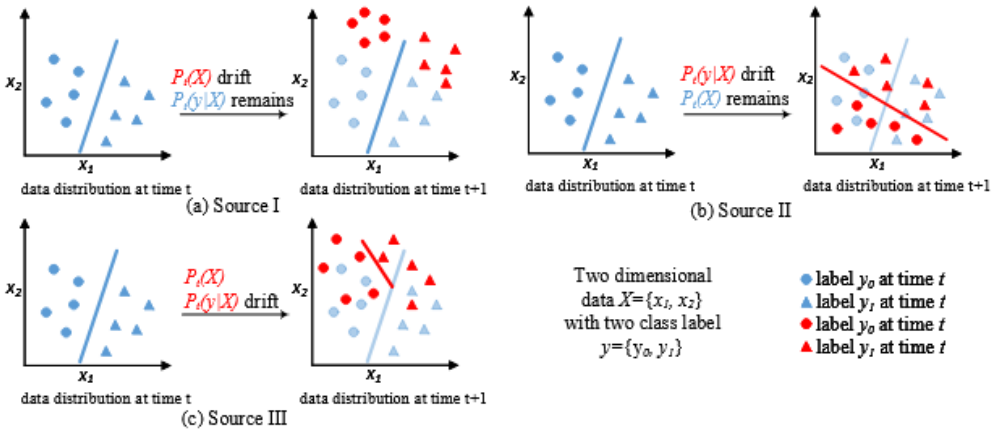
Ren等人提出CALIPER，一个与检测器、模型无关的纯数据驱动方法，用来估计后漂移数据量的充足性。他们假设特征空间中距离近的点，其下一步的演化也相似，并利用它进行回归预测，根据误差趋势判断重训练时机

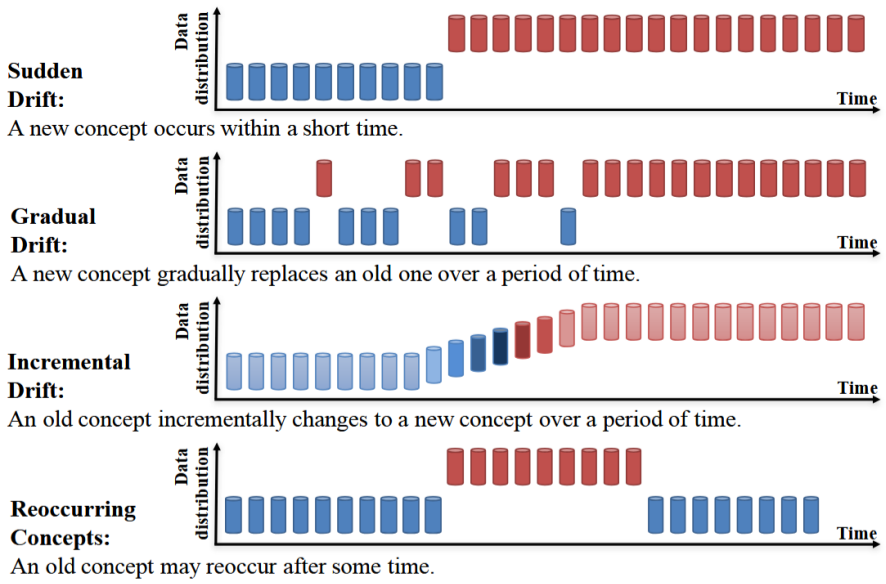
知识基础 数据漂移

- 数据漂移
  - 指数据流的统计特性随时间发生不可预见的变化

$$P_t(X,y) \neq P_{t+1}(X,y)$$

  - 由 $P(X,y) = P(X) \times P(y \mid X)$，漂移可以来自：
    - 特征分布变化（$P(X)$变化）：输入数据的分布变了，但分类边界没变
    - 决策边界变化（$P(y \mid X)$ 变化）：分类规则变了，模型准确率下降
    - 两者同时变化：最常见的情况
  - 常见的漂移类型
    - 突变漂移，概念在某一时刻瞬间变化
    - 渐进漂移，概念在一段时间内逐渐变化
    - 增量漂移，概念逐步演变，每一步变化很小
    - 循环漂移，旧概念在未来重新出现





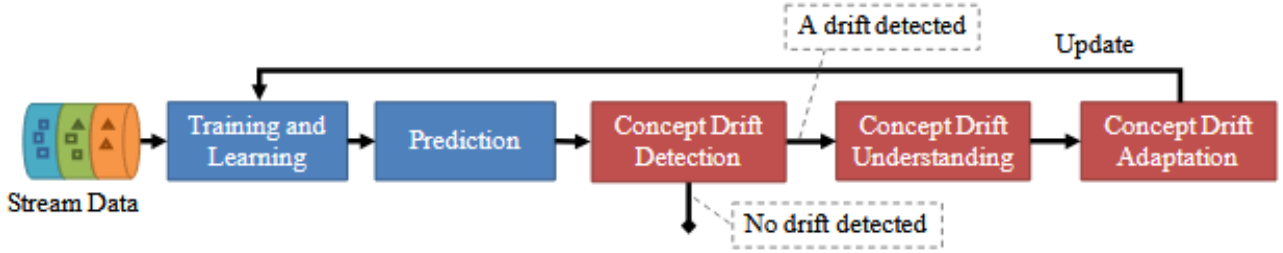
知识基础 流式数据

- 流式数据vs时序数据

数据类型	核心定义	对数据点间关系的假设	主要处理模式	对分布变化的要求	主要应用领域
时序数据	按时间顺序排列的观测值序列，存在自相关、趋势、季节性或周期性	关注相邻点或滞后点之间的统计关系，不能随意打乱顺序	批处理建模、滚动预测、递推预测	常要求平稳性或经过差分、去趋势等处理后近似平稳	能源负荷预测、气候建模、生物信号分析、设备状态预测
流式数据	连续、快速、潜在无限到达的数据序列，实时到达	样本间依赖弱，重点研究连续到达和分布变化	增量学习、滑动窗口更新、先预测后学习处理	需要检测、理解和适应分布变化	实时欺诈检测、在线推荐、广告点击率预测、传感器网络监控

知识基础 数据漂移

- 流程框架
  - 传统的机器学习有两个主要组成部分：
    - 训练和学习、预测
  - 概念漂移下的学习研究提出了三个新的组成部分：
    - 漂移检测：是否发生漂移
    - 漂移理解：何时、如何、在哪里发生
    - 漂移适应：对漂移存在的反应



算法原理 PUDD

Early Concept Drift Detection via Prediction Uncertainty

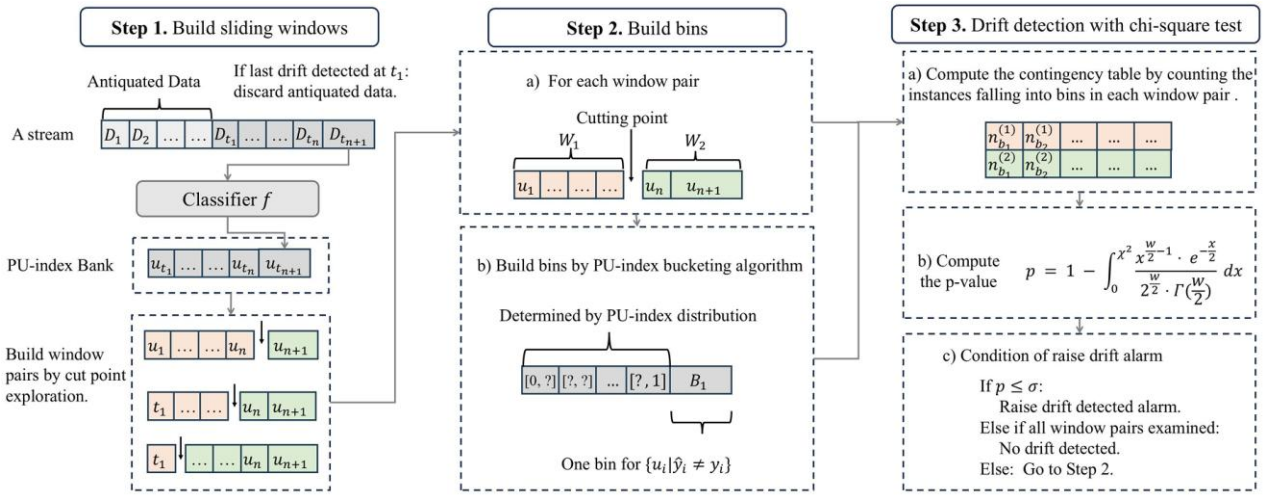
算法原理 TIPO

T	目标	在流式分类任务中实现早期检测漂移，提升检测灵敏度并保持模型性能稳定
I	输入	分类模型输出概率、流式分类数据集（表格*3、合成*3、图像*1）
P	处理	1.使用滑动窗口策略，丢弃过期数据 2.按正确、错误分类划分，构建直方图 3.检验计算，判断是否发生漂移
O	输出	是否发生概念漂移，最优切分点位置

P	问题	传统分类模型在流式场景下基于错误率的漂移检测不敏感，在漂移早期（分布变化但准确率未变）无法检测，导致检测延迟、漏检
C	条件	可及时获得数据真实标签、分类器可以输出概率
D	难点	如何构造更敏感的检测信号，保证在流式环境中准确性
L	水平	AAAI 2025 (CCF A)

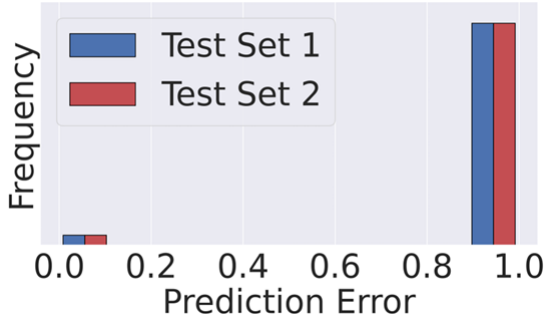
算法原理

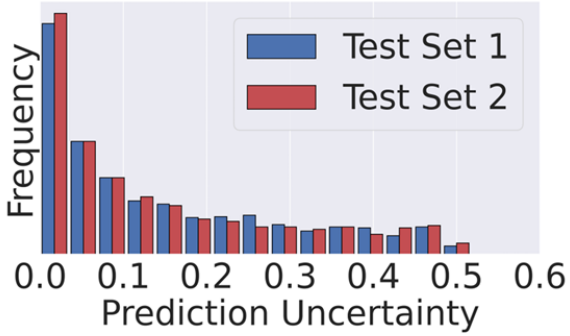
- 算法原理图
  - 构建滑动窗口
    - 先丢弃上一次漂移之前的过期数据，只保留最近的数据流
    - 枚举不同切分点，把当前数据流划分成两个窗口
  - 基于PU-index构建分桶
    - 对每个样本计算预测不确定性指数，再根据PU-index分布构造直方图分桶
  - 使用卡方检验判断漂移
    - 对两个窗口的PU-index分布建立列联表，计算卡方检验的p-value
    - 如果小于阈值，触发漂移警告



算法原理

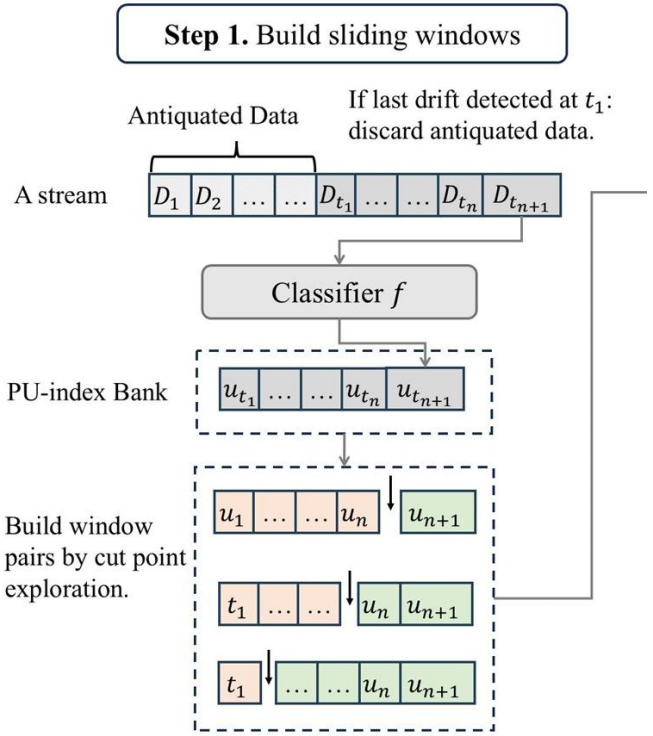
- 创新点
  - 把漂移检测信号从错误率换成PU-index
    - 错误率是0/1信号，只能告诉我们模型是否预测错误
    - PU-index是连续信号，可以捕捉早期漂移的状态
  - 理论上证明PU-index比错误率更敏感
    - 如果PU-index分布没有变化，那么错误率也不会变化
    - 即使错误率不变，PU-index分布仍可能变化
  - 设计了自适应分桶方法算法
    - PU-index是连续值，不能直接做卡方检验
    - 把PU-index分布转化为直方图，同时尽量满足卡方检验对样本频数的要求
    - 既保留不确定性分布变化，又避免检测过于敏感导致误报





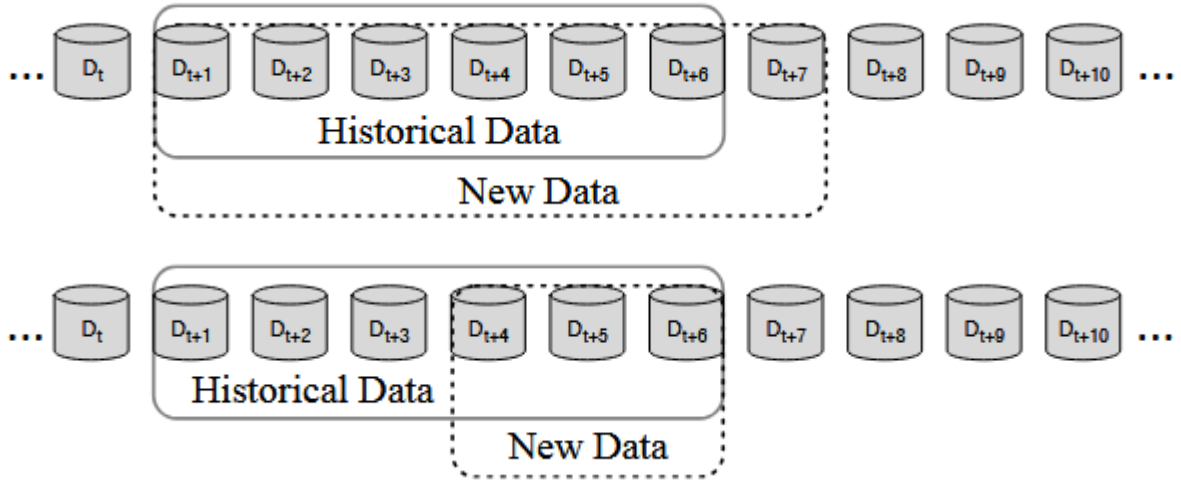
算法原理 构建滑动窗口

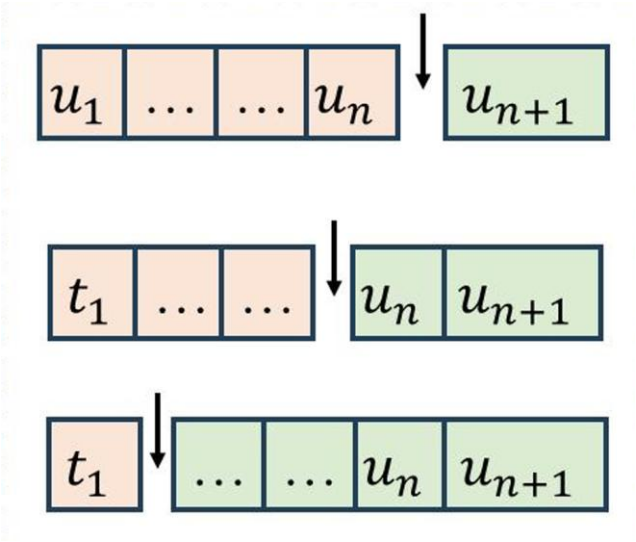
- 模块概述
  - 目标：漂移发生位置是未知的，使用两个滑动窗口准确定位漂移发生位置，避免之后进行的分布比较失真
  - 输入：
    - 当前保留的数据子流、分类器
  - 操作：
    - 丢弃漂移前的过期数据
    - 在当前子流中枚举所有可能切分点、并划分为两个窗口
    - 每个窗口对进入后续统计检测流程
  - 输出：
    - 一组候选窗口对



算法原理 构建滑动窗口

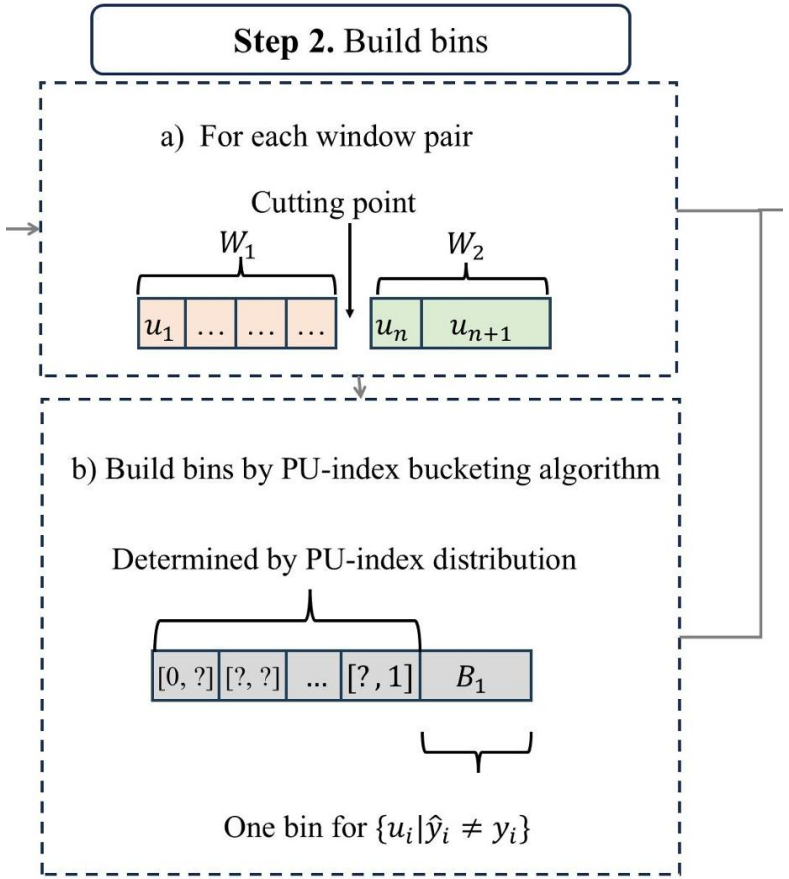
- 丢弃过期数据
  - 作用：确保后续比较是在当前分布附近进行，而不会被早期分布干扰
  - 从上次发生漂移的时间点t_1开始，将之前的数据D_{1,t_1}丢弃，保留$D_{t_1,t}$
- 枚举切分点
  - 对于子流中的每一个位置r，都将数据划分为两部分
  - 一部分代表较早阶段的数据，另一部分代表较新的数据





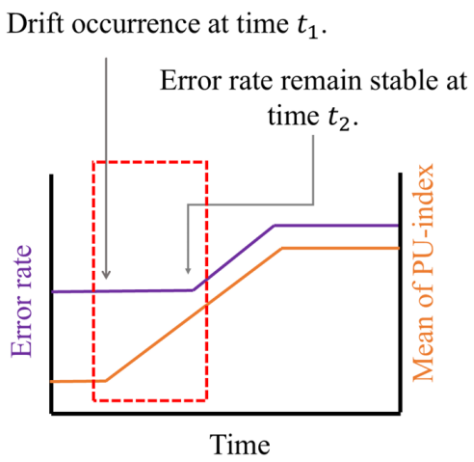
算法原理 基于PU-index构建分桶

- 模块概述
  - 目标：构建一个比错误率更敏感的漂移检测信号表示形式，并将其转化为可用于统计检验的分布结构
  - 输入：
    - 当前两个窗口的数据$D_{t_1,r}$和$D_{r,t}$
    - 分类器输出的预测概率$f(x)$
    - 每个样本的真实标签y_i
  - 操作：
    - 计算每个样本的PU-index
    - 对正确分类样本的PU-index进行自适应分桶
  - 输出：
    - 两个窗口对应的PU-index直方图分布

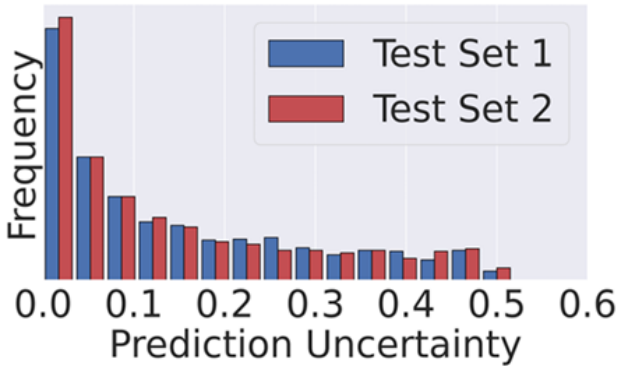


算法原理 基于PU-index构建分桶

- 预测不确定性指数
  - 对于样本中的已分类数据，定义PU-index为$u_i = 1 - f_{y_i}(x_i)$
  - 优点
    - 相比于错误率保留更多信息，可以衡量模型对真实标签的不确定性
    - 和分类任务目标直接相关，原始特征分布可能高维、噪声大



- 自适应分桶
  - 先在第一个窗口的正确分类上构建，用同一套去统计第二个窗口
  - 正确分类样本：PU-index的变化最有价值
    - 虽然分类正确，但它们的置信度可能已经发生变化
    - 早期漂移信号主要藏在正确分类样本内部
  - 错误分类样本：统一进入一个桶
    - 样本数量少，分桶后容易稀疏，且分桶收益不明显



算法原理 卡方检验判断漂移

- 模块概述
  - 目标：利用两个窗口的PU-index分布，进行统计显著性检验并输出是否发生漂移
  - 输入：
    - 两个窗口的PU-index分桶结果
  - 操作：
    - 将分桶结果转化为列联表
    - 根据列联表计算期望频数、卡方统计量
    - 取最小p_value和阈值比较
  - 输出：
    - 是否发生概念漂移、最优切分点

Step 3. Drift detection with chi-square test

a) Compute the contingency table by counting the instances falling into bins in each window pair .

$n_{b_1}^{(1)}$	$n_{b_2}^{(1)}$	...	...	...
$n_{b_1}^{(2)}$	$n_{b_2}^{(2)}$	...	...	...

b) Compute the p-value

$$p = 1 - \int_0^{\chi^2} \frac{x^{\frac{w}{2}-1} \cdot e^{-\frac{x}{2}}}{2^{\frac{w}{2}} \cdot \Gamma(\frac{w}{2})} dx$$

c) Condition of raise drift alarm

If $p \leq \sigma$:
Raise drift detected alarm.
Else if all window pairs examined:
No drift detected.
Else: Go to Step 2.

实验设计

- 数据资源
  - 表格数据：airline、 elec2、powersupply、sine、mixed、sea variants
  - 图像数据：CIFAR-10-CD
- 分类模型
  - 深度神经网络（DNN）、高斯朴素贝叶斯（GNB）、流式决策树（VFDT）
- 对比方法
  - 错误率基准方法：DDM、EDDM、ADWIN、HDDM、PH
  - 分布基准方法：KSWIN(2020)、MCDD(2024)
  - 适应方法：AMF(2021)、 IWE(2022)、NS(2021) 、ADLTER(2022)、MCD-DD（2024）
- 指标：分类准确率

实验设计 对比实验

- 实验结果

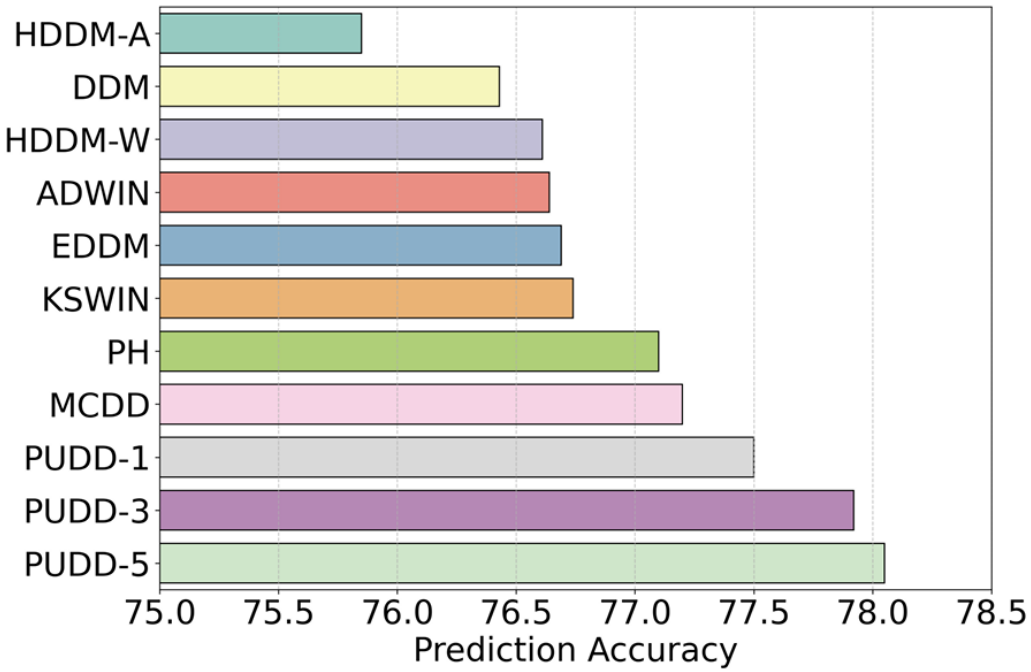
ddm name	airline	elec2	mixed	ps	sea0	sine
AMF	38.56	66.24	49.49	69.63	93.67	49.52
IWE	38.02	68.90	49.47	64.10	93.14	49.51
NS	**67.91**	76.42	81.09	72.39	93.54	91.01
ADLTER	**70.00**	76.10	87.63	72.48	93.40	**92.18**
MCD-DD	63.65	69.81	86.68	71.66	97.66	90.21
PUDD-1	63.78	**77.28**	**89.51**	**72.68**	**98.47**	**94.52**
PUDD-3	**64.62**	**76.77**	**89.47**	**72.79**	**98.44**	**94.76**
PUDD-5	64.45	**76.92**	**89.37**	**72.74**	**98.49**	90.90

- 分析
  - PUDD在多数情况下进入前三
  - PUDD对不同分类模型的适配性不同
  - 阈值的选择对性能有影响，但一般不起决定性作用

		Incremental Training						Training only at Initialization or Adaptation					
Classifier	ddm name	airline-I	elec2-I	mixed-I	ps-I	sea0-I	sine-I	airline-O	elec2-O	mixed-O	ps-O	sea0-O	sine-O
DNN	ADWIN	61.65	71.94	78.45	71.12	97.70	79.27	59.36	69.71	84.41	65.95	95.18	86.59
	DDM	61.29	71.02	**80.72**	71.37	96.86	**86.41**	56.79	69.51	83.03	69.35	93.55	80.32
	EDDM	62.13	70.39	78.01	65.21	96.89	75.34	**61.17**	69.22	71.16	67.57	92.18	76.54
	HDDM-A	62.70	71.16	76.70	68.97	97.84	81.65	59.13	68.98	84.27	67.81	95.11	**86.82**
	HDDM-W	61.50	71.23	77.07	68.74	97.84	**85.43**	**62.42**	68.48	84.32	66.89	92.24	86.57
	KSWIN	63.02	70.56	78.58	70.80	97.88	79.00	**61.34**	69.08	**84.43**	66.46	91.60	83.66
	PH	62.15	72.25	79.49	70.86	97.73	78.07	60.36	68.24	84.36	68.83	**95.21**	**86.88**
	PUDD-1	**63.31**	**74.92**	77.39	**72.25**	**97.94**	**86.19**	60.90	69.35	82.65	**71.47**	94.89	83.39
	PUDD-3	**63.21**	**74.93**	**80.05**	**72.23**	**98.04**	85.12	60.16	68.98	**84.65**	**70.37**	**95.99**	84.97
	PUDD-5	**63.35**	**74.92**	**82.81**	**72.24**	**98.23**	82.51	60.19	68.68	**84.90**	**70.20**	**96.29**	85.09
GNB	ADWIN	50.17	68.90	**83.95**	70.06	94.18	82.49	**54.66**	**68.30**	83.62	68.78	93.93	82.45
	DDM	52.94	67.75	83.82	69.63	93.97	82.07	52.43	67.60	**83.59**	67.52	93.48	81.90
	EDDM	**62.72**	67.73	83.19	70.04	94.06	83.12	54.11	67.64	74.65	**70.04**	**94.06**	73.88
	HDDM-A	52.80	67.73	83.92	70.87	94.28	83.25	**55.62**	**67.73**	83.66	**71.24**	93.96	**83.36**
	HDDM-W	48.66	67.73	83.91	71.06	92.11	82.83	48.62	67.71	83.60	69.53	91.63	83.25
	KSWIN	49.84	67.87	83.92	71.23	91.72	81.88	48.83	67.63	83.59	67.99	90.02	81.32
	PH	49.35	70.12	83.88	70.36	94.34	**83.54**	49.02	**70.04**	83.61	68.67	94.12	83.10
	PUDD-1	**53.57**	**70.85**	82.99	**71.88**	**94.61**	83.12	51.05	62.76	79.23	**71.13**	**94.25**	81.48
	PUDD-3	**53.03**	**70.85**	**83.92**	**71.59**	**94.81**	**83.39**	49.45	59.32	83.58	**71.20**	**94.60**	**83.80**
	PUDD-5	52.16	**70.69**	**84.12**	**71.59**	**94.85**	**83.43**	**54.37**	59.44	**83.96**	70.40	**94.62**	**83.38**
VFDT	ADWIN	60.39	**73.98**	84.30	71.69	94.77	87.11	61.22	**74.29**	83.54	68.78	92.96	**85.74**
	DDM	60.16	**74.82**	84.14	70.68	94.86	86.50	59.28	**74.75**	82.31	67.53	93.63	82.23
	EDDM	61.19	73.81	83.15	70.06	94.17	85.59	**62.31**	73.81	73.73	70.06	93.60	77.72
	HDDM-A	60.95	73.90	**84.40**	70.84	**95.20**	**87.53**	60.29	73.83	**83.66**	**71.24**	93.93	85.21
	HDDM-W	61.11	73.80	**84.40**	70.99	93.50	**87.45**	**61.92**	73.73	83.59	69.54	91.76	85.28
	KSWIN	**61.30**	**74.10**	**84.42**	71.27	93.40	86.22	**62.06**	**74.10**	83.57	67.51	89.84	82.60
	PH	60.95	73.70	83.56	70.88	94.69	87.15	60.97	73.99	83.30	68.67	93.85	85.19
	PUDD-1	61.38	73.86	84.25	**71.77**	95.13	87.33	61.16	69.79	82.13	**71.13**	**94.10**	82.21
	PUDD-3	**61.57**	73.84	83.94	**71.79**	**95.21**	87.42	59.90	69.79	84.04	**71.20**	**94.63**	**85.81**
	PUDD-5	**61.57**	73.64	84.01	**71.79**	**95.24**	**87.63**	57.04	71.83	**84.16**	70.40	**94.56**	**86.01**

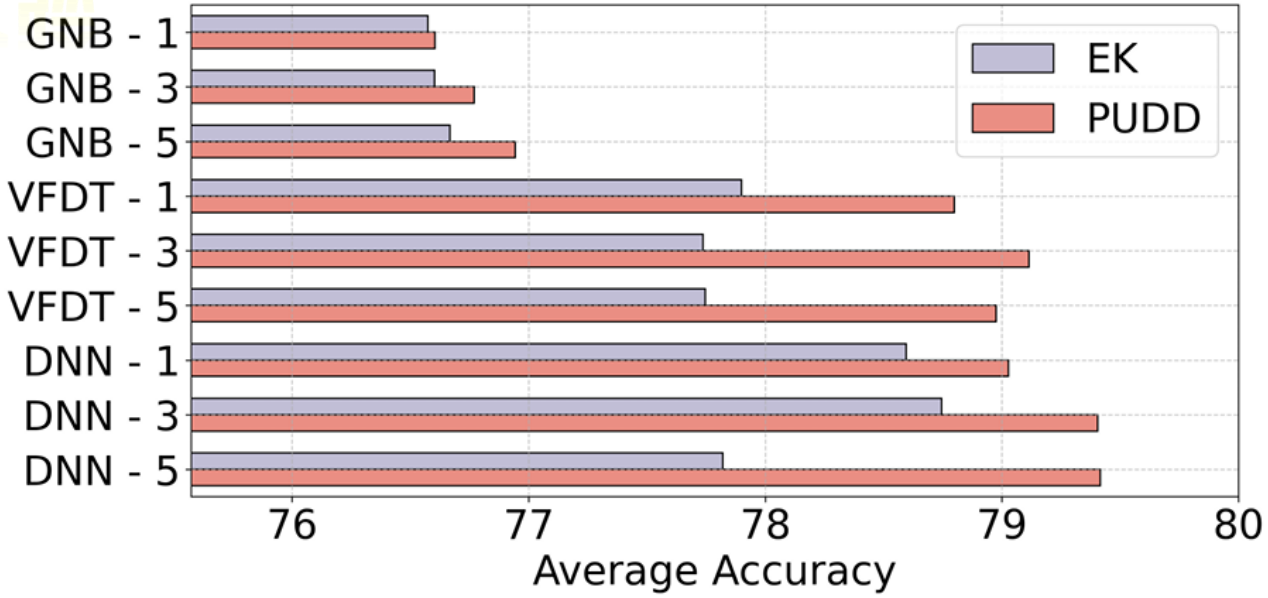
实验设计 对比实验

- 实验结果
- 分析
  - PUDD 不只适用于低维表格流，也能用于高维图像数据
  - PUDD 没有直接在图像像素空间做分布比较，而是在分类器输出概率上做PU-index检测，进而绕开了高维图像分布检测困难的问题



实验设计 消融实验

- 实验结果



- 消融情况
  - EK(Ei-kMeans)采用传统空间分桶方法
  - DNN提升最明显
    - DNN 输出概率分布更丰富

算法原理 CALIPER

When to Retrain after Drift:
A Data-Only Test of Post-Drift Data Size Sufficiency

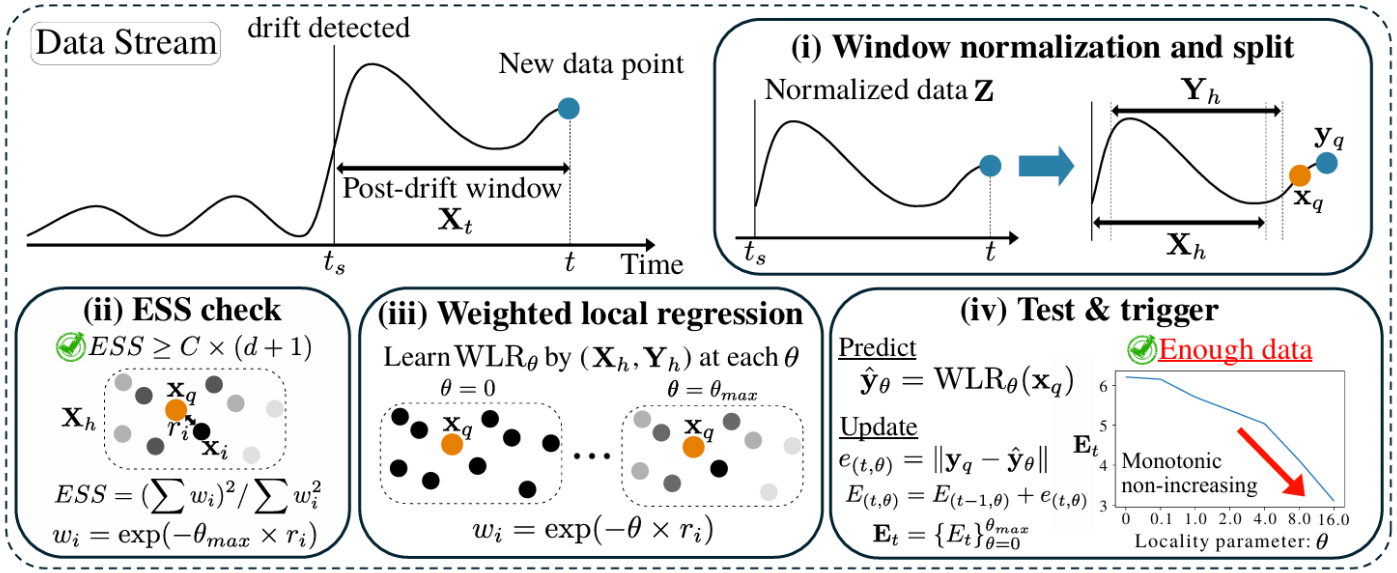
算法原理 TIPO

T	目标	能够在漂移后，自动判断何时积累了足够数据并触发稳定重训练
I	输入	漂移报警后的流数据，数据集(图像-MoCap，表格–TEP、Automobile，合成-Dysts)，外部漂移检测器
P	处理	1.对漂移窗口数据进行标准化，并拆分为一步预测参考对与当前查询对 2.通过不同参数进行加权局部回归，同时利用有效样本数ESS约束 3.检查代理误差， 当满足ESS与单调性条件时，触发下游模型重训练
O	输出	漂移后是否可以重训练，估计出的最小重训练数据量

P	问题	固定窗口策略容易过早或过晚重训练，导致过拟合、性能退化或恢复延迟
C	条件	数据流已经由外部检测器检测到漂移，数据持续到达，下游预测模型作为黑盒，不依赖漂移后测试标签
D	难点	如何仅利用漂移后的数据本身，判断新概念是否已经形成足够稳定的局部状态依赖，从而避免过早重训练或过晚重训练
L	水平	ICLR 2026（CCF A）

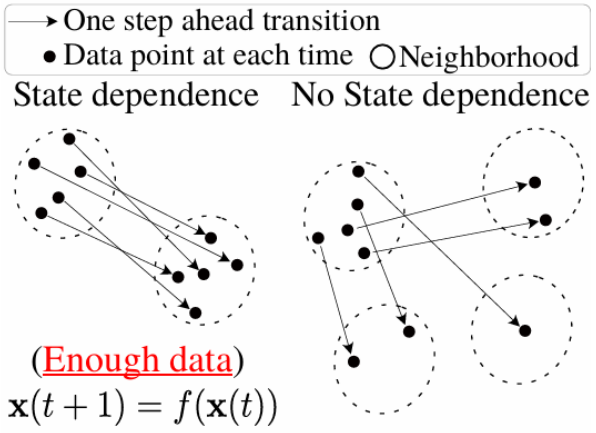
算法原理

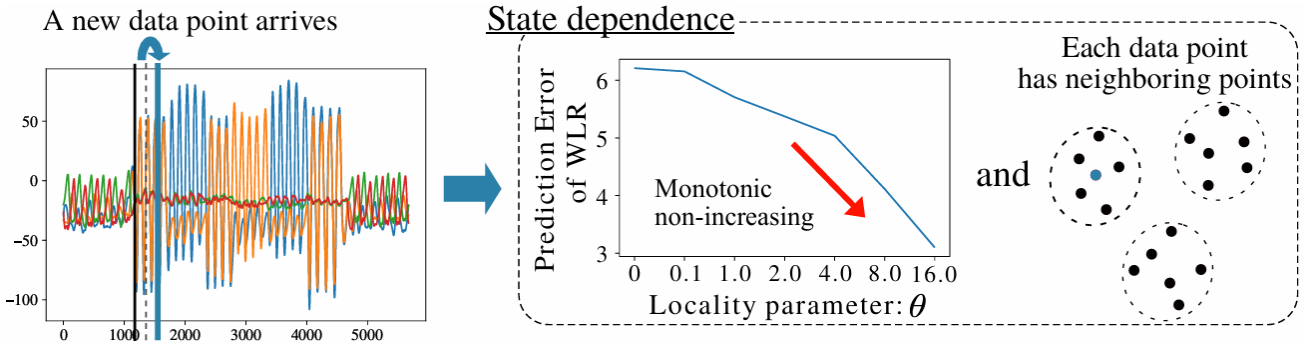
- 算法原理图
  - 窗口归一化与分割
    - 对当前数据窗口进行标准化处理，并用历史点预测下一时刻
  - ESS检查
    - 在局部的邻域尺度下计算有效样本数
  - 加权局部回归
    - 在不同局部性参数下，对邻近样本进行加权回归来预测当前点
    - 计算对应的预测误差
  - 测试与触发
    - 判断预测误差是否随局部性增强而单调不增
    - 若满足条件，则认为数据足够，触发模型重训练



算法原理

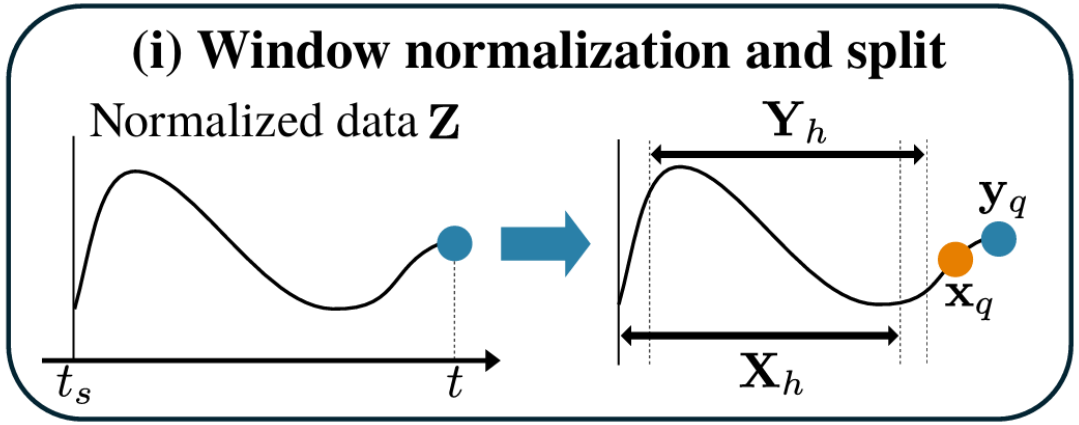
- 创新点
  - 问题建模的改变
    - 使问题从检测问题变成决策问题
    - 完全围绕数据本身展开，不依赖模型内部信息或标签
  - 局部状态依赖
    - 如果偏移后更局部的邻域能带来更低的一步预测误差，并且邻域有效样本数足够，就认为当前数据量足以稳定重训
    - 利用有效样本数（ESS）约束来避免样本稀疏导致的错误判断





算法原理 窗口归一化与分割

- 模块概述
  - 目标：
    - 将原始的偏移后数据窗口转化，为后续操作提供统一、可比较的数据表示分类
  - 输入：
    - 漂移发生后累积得到的时间序列窗口
  - 操作：
    - 对窗口内数据进行标准化处理
    - 将连续时间序列重构为一步预测样本对
  - 输出
    - 历史样本对
    - 当前查询点

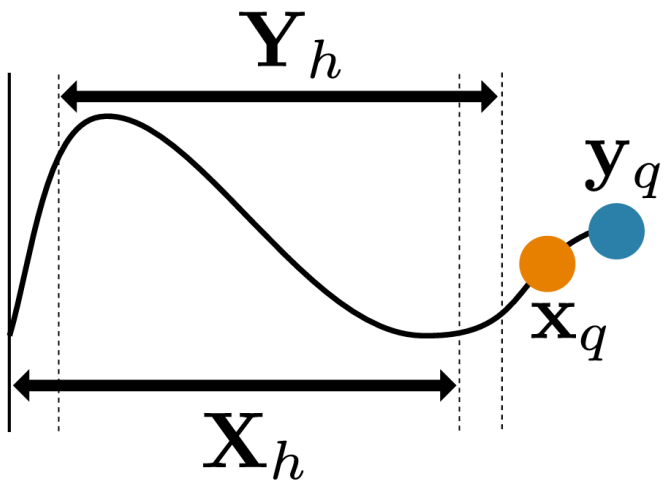


算法原理 窗口归一化与分割

- 窗口归一化
  - 原始数据流中，不同时间段的数据尺度可能完全不同
  - 保证后续距离、邻域、局部性这些概念是有意义的
- 一步预测
  - 用当前状态预测下一步状态，判断数据是否已经形成稳定的动态规律

    X_h：历史输入（过去状态）　Y_h：历史输出（下一步状态）

    x_q：当前状态（query）　y_q：当前真实下一步
  - 原始数据$x(t_s), x(t_s+1), \dots, x(t)$
  - 重构数据$(x_1 \to x_2), (x_2 \to x_3), \dots, (x_{n-1} \to x_n)$

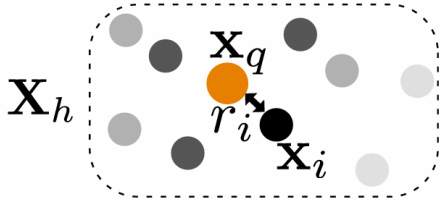


算法原理 ESS检查

- 模块概述
  - 目标：
    - 确保在进行局部建模时，参与计算的有效样本数量足够，避免因为邻域过小或权重过于集中导致的偶然拟合和错误判断
  - 输入：
    - 历史样本对
    - 当前查询点
  - 操作：
    - 根据加权样本计算有效样本数
    - 与阈值$C(d+1)$进行比较
  - 输出
    - 判断当前窗口在局部尺度下是否具有足够的有效样本数

(ii) ESS check

$ESS \geq C \times (d+1)$

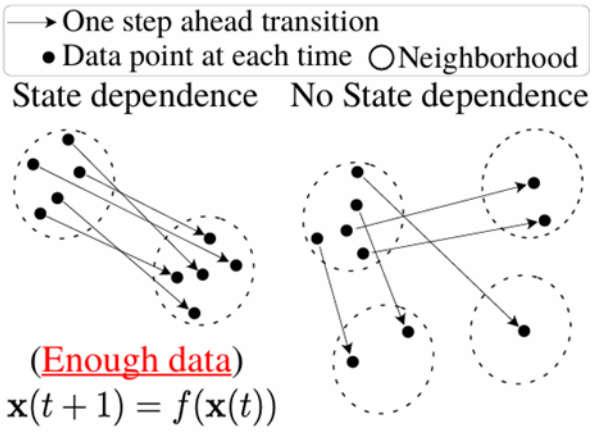


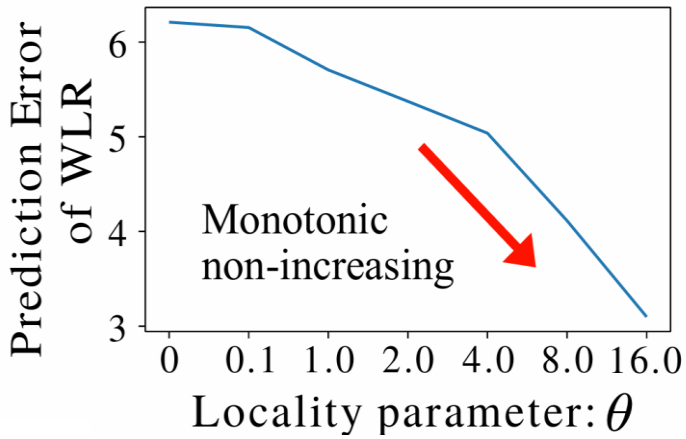
$$ESS = (\sum w_i)^2 / \sum w_i^2$$

$$w_i = \exp(-\theta_{max} \times r_i)$$

算法原理 ESS检查

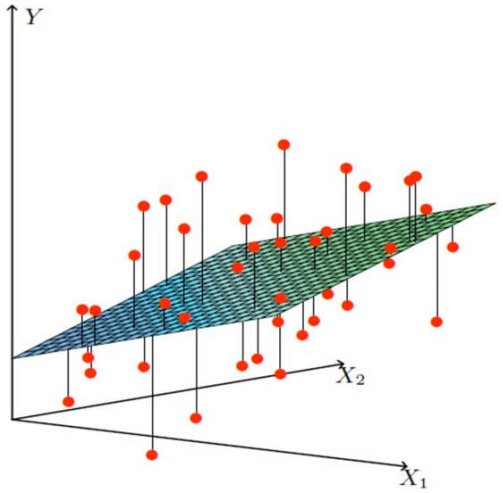
- 错误触发
  - 如果邻域过小（也就是局部性参数θ很大）
    - 误差看起来很小，因为只是过拟合了几个点
- 有效样本数(ESS)
  - 衡量在加权情况下，参与建模的等效样本数量
  - $w_i(\theta) = \exp(-\theta \cdot r_i)$，其中$r$是距离，$w_i$是权重
  - $ESS = \frac{(\sum_i w_i)^2}{\sum_i w_i^2}$
    - 所有样本权重差不多：ESS 接近样本总数
    - 只有少数样本权重大：ESS 会很小
    - 只在最大局部性下检查 ESS，最极端的局部情况满足，其他情况都满足





算法原理 ESS检查

- 阈值$C(d+1)$
  - d 是数据维度，+1：对应偏置项
    - 局部回归模型不仅拟合斜率，还要拟合截距
  - 有效样本数必须至少大于模型参数个数，并且具有一定冗余
  - 参数C
    - 有效样本数要明显多于参数数目
    - 降低方差，提高数值稳定性，再次避免局部过拟合

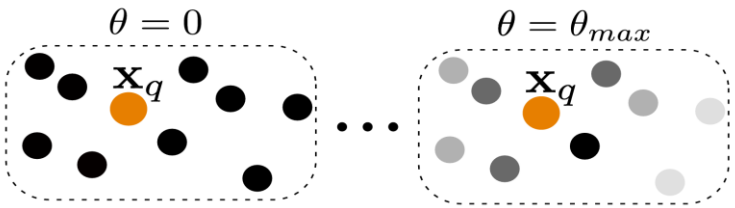


算法原理 加权局部回归

- 模块概述
  - 目标：
    - 在不同局部尺度下，对当前数据的预测能力进行估计，构造一个能够反映数据稳定性的误差信号
  - 输入：
    - 历史样本对(X_h, Y_h)
    - 当前查询点x_q及其真实下一步y_q
  - 操作：
    - 在每个θ下，用加权最小二乘拟合一个局部线性模型，并用该模型预测当前点的下一步，然后计算预测误差
  - 输出
    - 对每个θ，得到一个预测值$\hat{y}_\theta$和对应误差$e(t,\theta)$，并逐步累计为$E(t,\theta)$

(iii) Weighted local regression

Learn WLR_θ by $(\mathbf{X}_h, \mathbf{Y}_h)$ at each θ



$$w_i = \exp(-\theta \times r_i)$$

算法原理 加权局部回归

- 加权拟合
  - 线性模型的优势
    - 计算快、易解释、在小邻域内近似合理

$$\min_{\beta} \sum_{i} w_i(\theta) \| Y_h^{(i)} - X_h^{(i)} \beta \|^2$$

  - 用拟合后的函数去预测$\hat{y}_\theta = f_\theta(x_q)$
  - 获得误差并累加$e(t,\theta) = \| y_q - \hat{y}_\theta \|$，$E(t,\theta) = E(t-1,\theta) + e(t,\theta)$
    - 单次误差：噪声大、不稳定、容易误判
    - 累加的作用：平滑时间维度的波动，误差趋势更稳定，更适合做单调性判断

算法原理 测试与触发

- 模块概述
  - 目标：
    - 在不同局部尺度下，对当前数据的预测能力进行估计，构造一个能够反映数据稳定性的误差信号
  - 输入：
    - 不同局部性参数下的累计误差$E(t,\theta_1), E(t,\theta_2), \ldots, E(t,\theta_k)$
    - 已通过 ESS 检查的数据窗口
  - 操作：
    - 检查误差序列是否满足随局部性增强单调不增
  - 输出
    - $R(X_t) = 1$：触发重训练
    - $R(X_t) = 0$：继续积累数据

(iv) Test & trigger

Predict

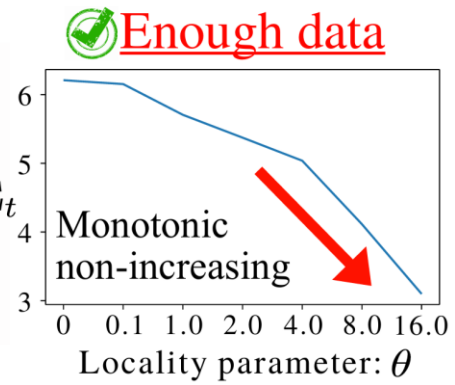
$$\hat{\mathbf{y}}_\theta = \mathrm{WLR}_\theta(\mathbf{x}_q)$$

Update

$$e_{(t,\theta)} = \|\mathbf{y}_q - \hat{\mathbf{y}}_\theta\|$$

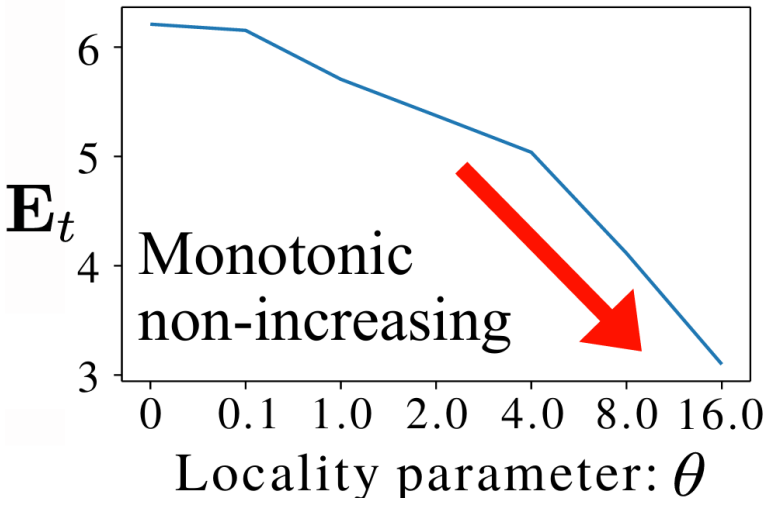
$$E_{(t,\theta)} = E_{(t-1,\theta)} + e_{(t,\theta)}$$

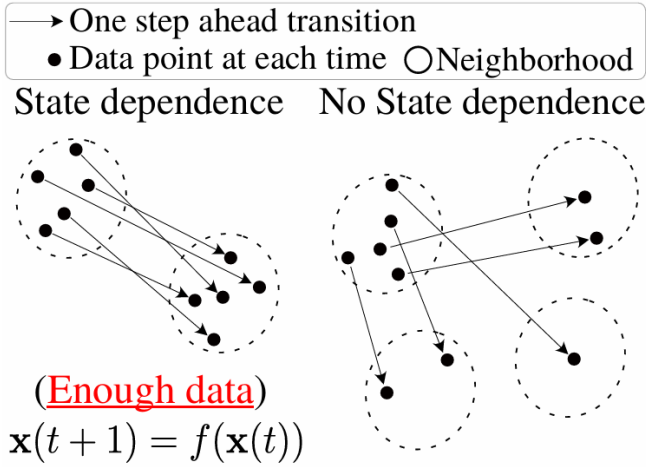
$$\mathbf{E}_t = \{E_t\}_{\theta=0}^{\theta_{max}}$$



算法原理 测试与触发

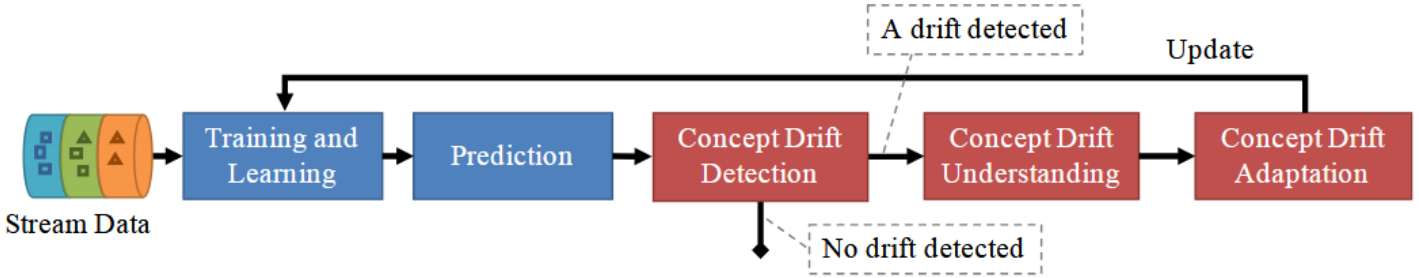
- 重训练判据
  - $E(t,\theta_1) \geq E(t,\theta_2) \geq \cdots \geq E(t,\theta_k)$
    - 随着θ增大（邻域越来越局部），预测误差不增加
  - 单调下降证明数据已经稳定
    - 数据不稳定：
      - 在漂移过渡期，局部结构混乱
    - 数据已经稳定：
      - 数据内部有规律，相似的数据有相似的未来状态





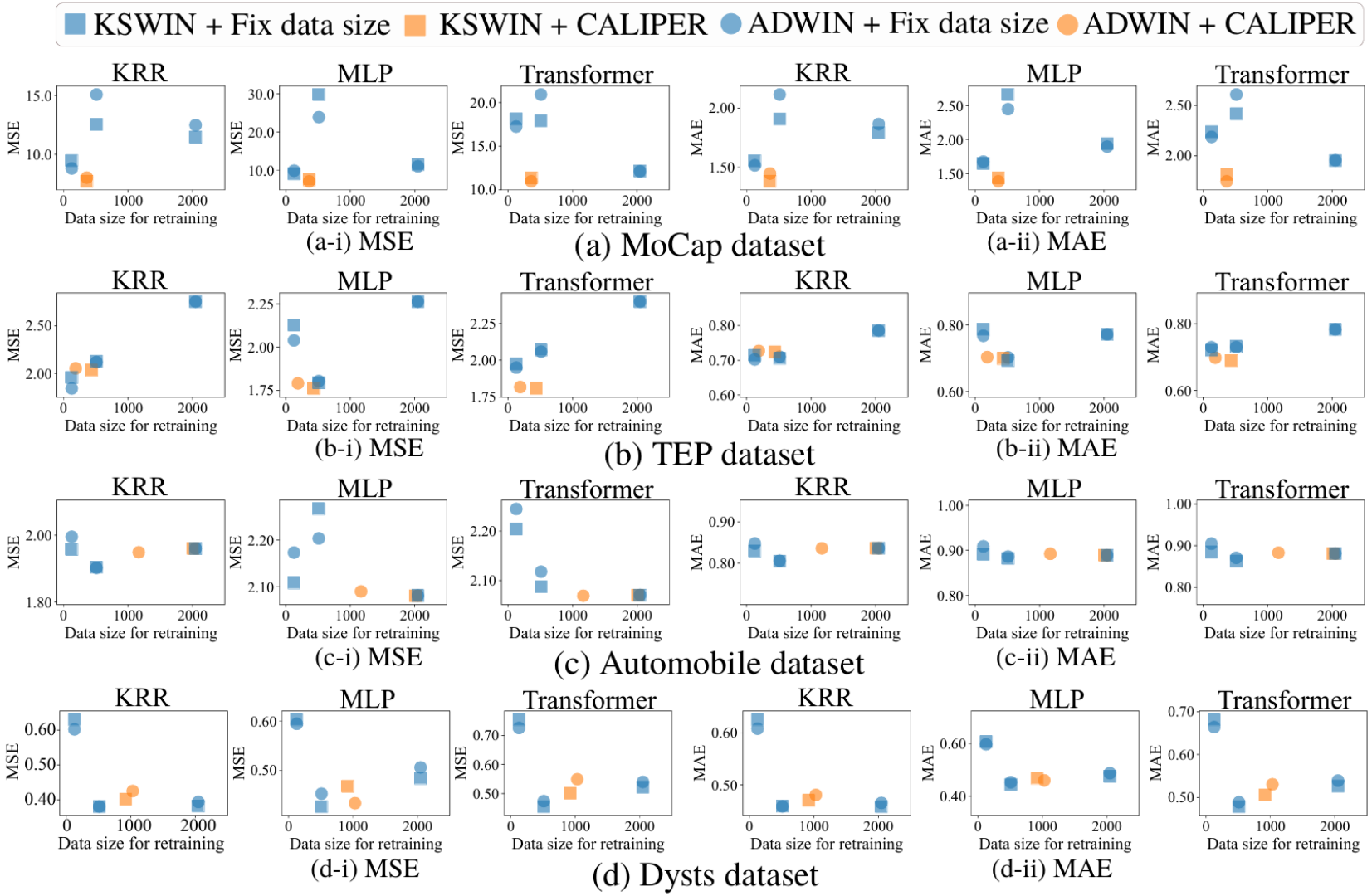
实验设计

- 数据资源
  - (图像-人体动作MoCap，表格-工业控制TEP、多传感器Automobile，合成-Dysts)
- 对比方法
  - 漂移检测器：ADWIN（2007）、KSWIN（2020）
  - 重训练方法：固定窗、增量学习
  - 下游模型：KRR、MLP、Transformer
- 指标：均方误差（MSE）、平均绝对误差（MAE）



实验设计 对比实验

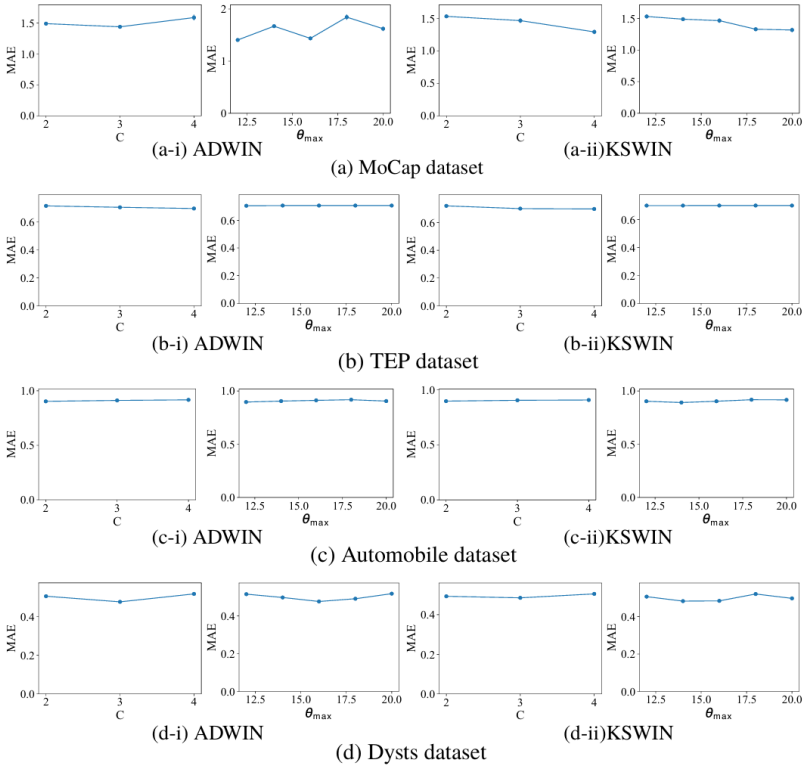
- 实验结果
- 分析
  - 误差曲线通常呈先降后升的趋势
  - CALIPER选的点通常落在误差曲线的低谷附近
  - 不同模型的曲线形状不同，但CALIPER在不同模型上仍然有效



实验设计 对比实验

- 实验结果

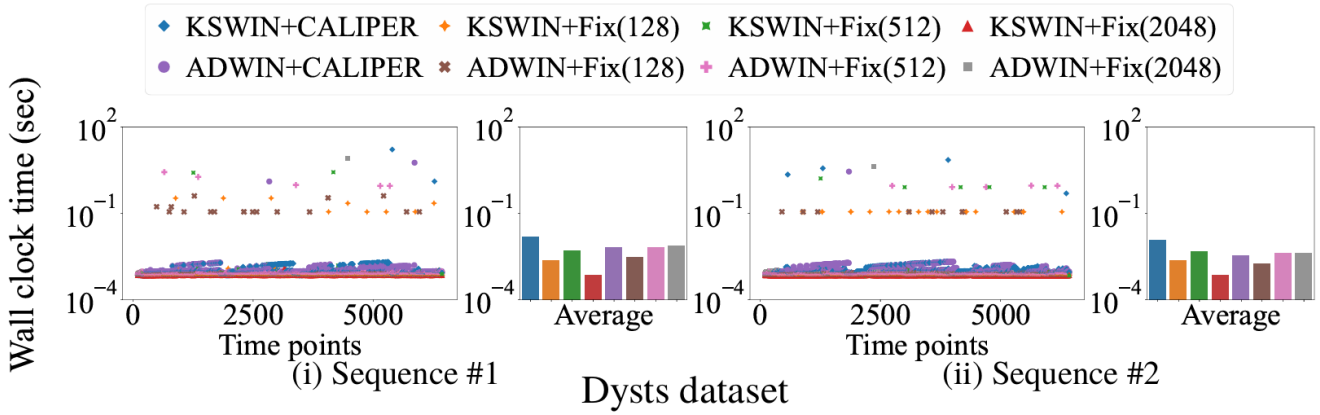
Model	Detector	MoCap		TEP		Automobile		Dysts	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
MLP	ADWIN	**7.106**	**1.387**	1.790	0.704	2.090	0.892	**0.432**	**0.460**
	KSWIN	7.449	1.436	**1.760**	**0.699**	**2.081**	**0.889**	0.467	0.468
	(Incremental)	412.6	8.462	3.699	0.879	2.406	0.894	71.75	1.524
Transformer	ADWIN	**10.96**	**1.744**	1.818	0.698	1.948	0.883	0.549	0.530
	KSWIN	11.35	1.810	**1.808**	**0.690**	2.070	0.881	**0.501**	**0.505**
	(Incremental)	22.08	2.654	2.376	0.767	**1.866**	**0.773**	0.582	0.585

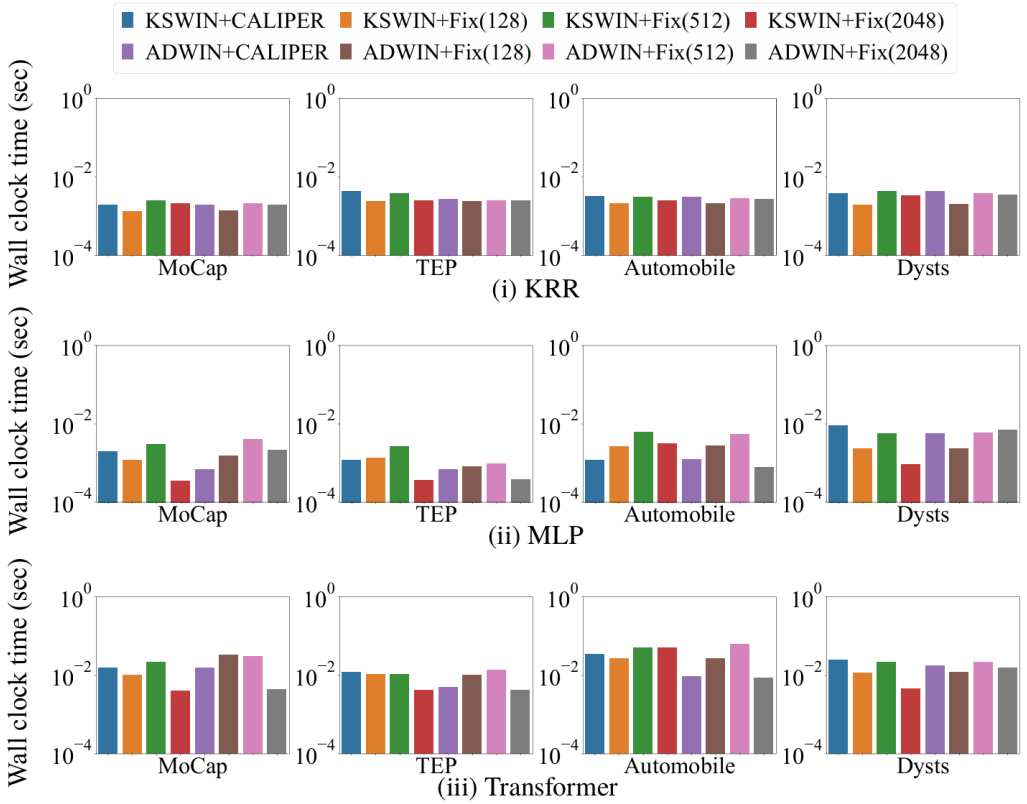


- 分析
  - CALIPER在大多数情况下优于增量学习
    - 在突发漂移下，梯度方向错误，增量学习更新会发散
    - CALIPER会等数据稳定了再学习
  - CALIPER对参数不敏感

实验设计 对比实验

- 实验结果





- 分析
  - CALIPER的开销和固定窗方法相近
  - 曲线平坦且均值较低，偶尔的峰值反映的是重新训练而非CALIPER

特点总结与未来展望

工作总结

特点总结与工作展望

- 特点总结

算法	所属阶段	核心思想	主要优势	主要不足
PUDD	漂移检测阶段	用预测不确定性替代传统错误率作为漂移信号	能发现错误率尚未变化的早期漂移	依赖预测概率质量，阈值仍需设置，标签延迟场景下使用受限
CALIPER	漂移后适应阶段	利用漂移后数据的状态依赖性判断数据是否足够	给出了计算漂移后稳定重训需要数据数量的模型	更适合突变漂移后的重训数据量判断，高维原始数据中需要良好表征

- 工作展望
  - 将早期漂移检测能力与重训时机判断机制结合
  - 面向高维、标签延迟等真实场景，提升漂移检测与恢复方法的稳定性、泛化性和实际部署能力

参考文献

[1] Lu P, Lu J, Liu A, et al. Early concept drift detection via prediction uncertainty[C]//Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence. 2025.

[2] Fujiwara R, Matsubara Y, Sakurai Y. When to retrain after drift: A data-only test of post-drift data size sufficiency[C]//Proceedings of the International Conference on Learning Representations. 2026.

道德经

知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者，寿。

谢谢！