

Anomaly Detection on Streaming Data

Three isolation-based algorithms, from unsupervised to semi-supervised

PhD Research Seminar

Jia (Justin) Liu

Department of Computer Science, University of Waikato
TAIAO programme, AI Institute

Supervisors

Prof. Albert Bifet
Prof. Bernhard Pfahringer
Dr. Guilherme Weigert Cassales
Dr. Fei Tony Liu

About me

Jia (Justin) Liu

PhD candidate
AI Institute
University of Waikato

jia.liu@waikato.ac.nz
jliu.me

Interests

Stream learning, anomaly
detection, label-efficient learning,
open-source ML software

2003 – 2007

B.Sc. Computer Science

Tianjin University of Science & Technology, China

2008 – 2018

Software engineer and team lead, China

QA, banking terminal software, a video server platform (team lead), the GBASE MPP database, and game server infrastructure

2019 – 2021

M.Sc. Computer Science, University of Waikato

Semi-supervised Active Learning for Image Classification, supervised by Prof. Eibe Frank

2021 – 2023

Research Programmer, AI Institute, TAI AO

Sole engineer of the TAI AO stream learning platform: Kafka and Flink, Kubernetes on AWS and GCP, live NZ weather radar feed

2023 – 2026

Ph.D. Computer Science, University of Waikato

Anomaly Detection on Streaming Data, TAI AO programme (MBIE), University of Waikato scholarship. Under examination

Outline

1

Background

Anomaly detection, Isolation Forest, data streams, research questions

2

Unsupervised methods

SiForest (reservoir sampling, distributional equivalence) and AIF (MTMM tree replacement)

3

Semi-supervised methods

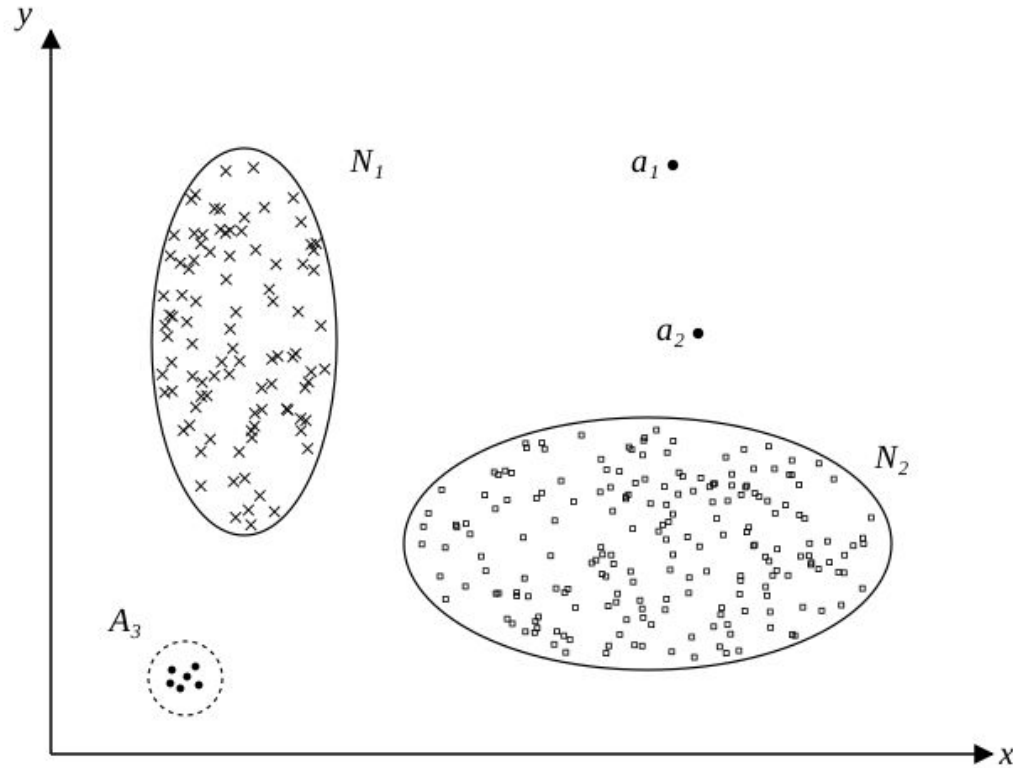
S³AIF: discrimination power with disagreement-based active learning

4

Synthesis

Answers to the research questions, cross-cutting findings, limitations, future work

Introduction to Anomaly Detection



N_1, N_2 : normal regions. a_1, a_2 : point anomalies. A_3 : a small anomalous cluster.

Anomalies are few and different.

Formal Definition

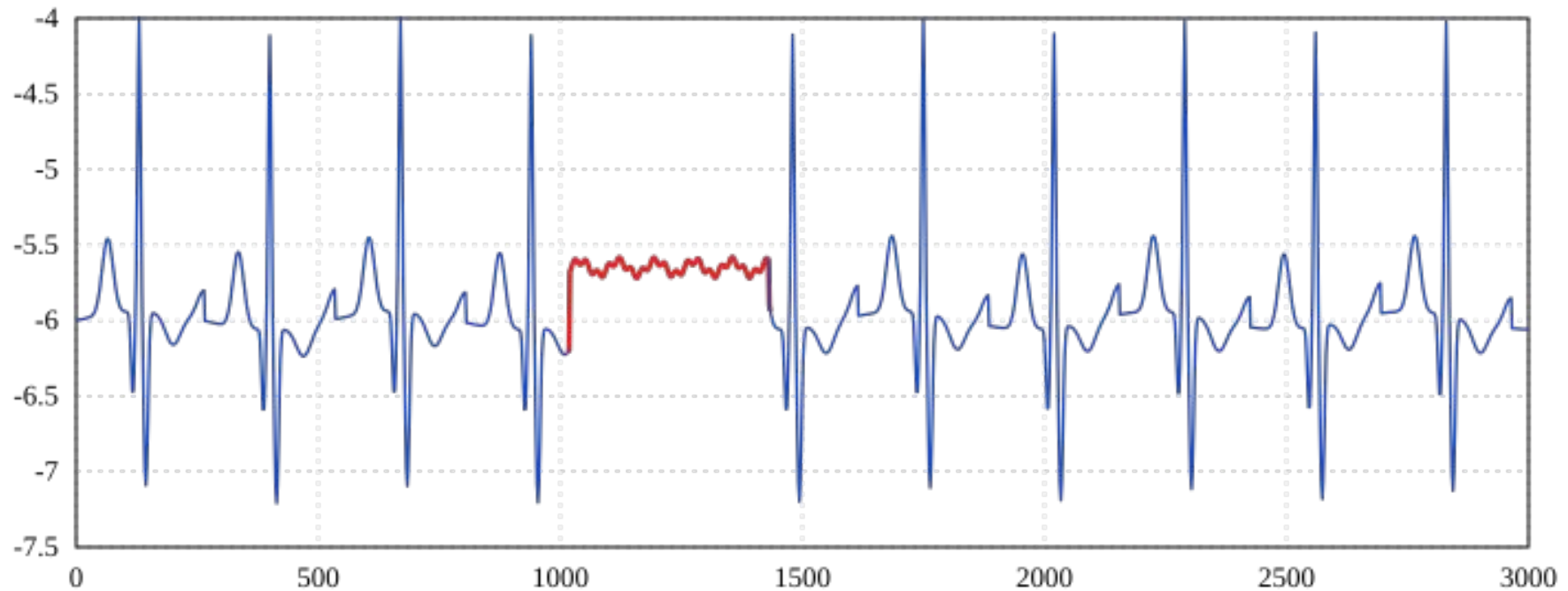
"Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior."

— Chandola, Banerjee, and Kumar 2009

Core Objective

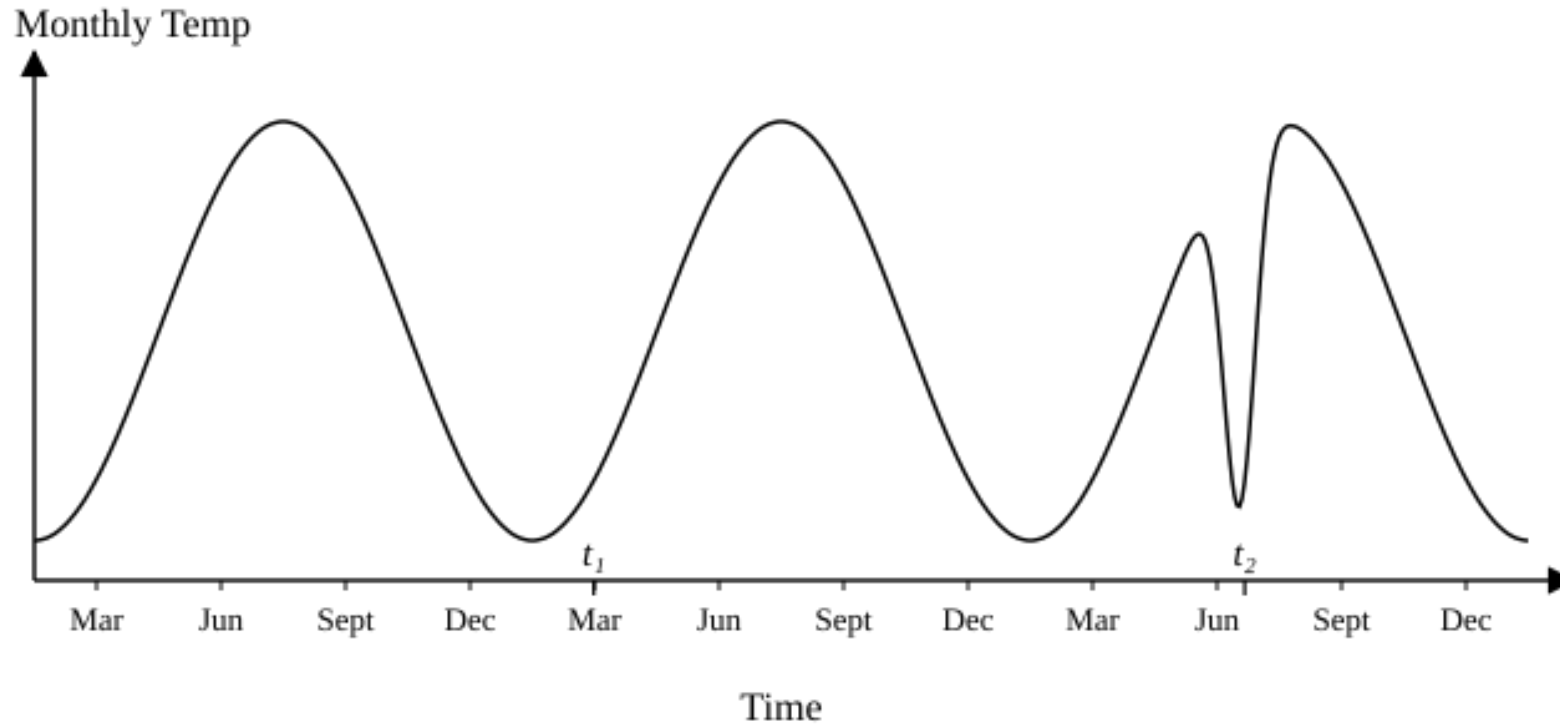
Anomaly detection is therefore the action to label anomalies and normal points correctly.

Collective Anomaly



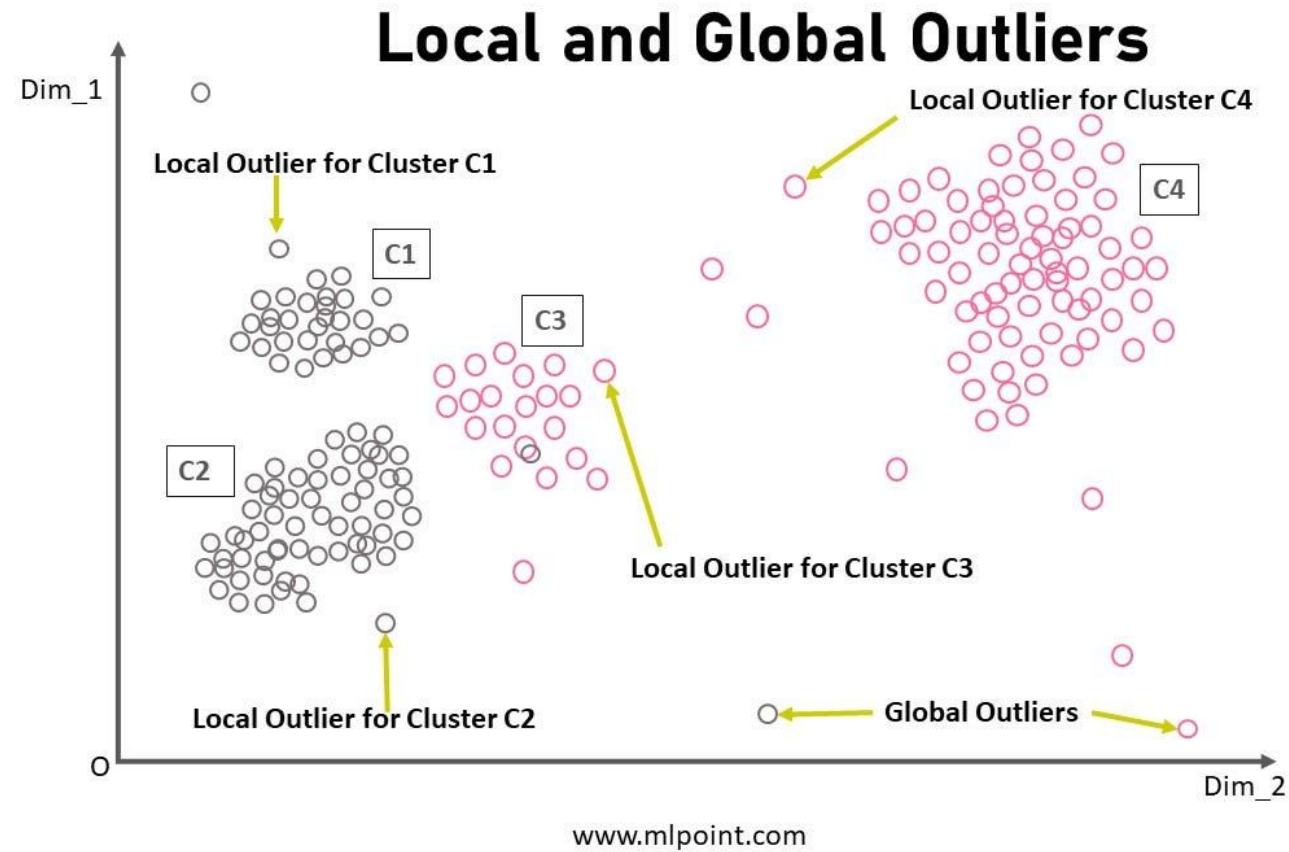
Abnormal Heartbeat

Contextual Anomaly

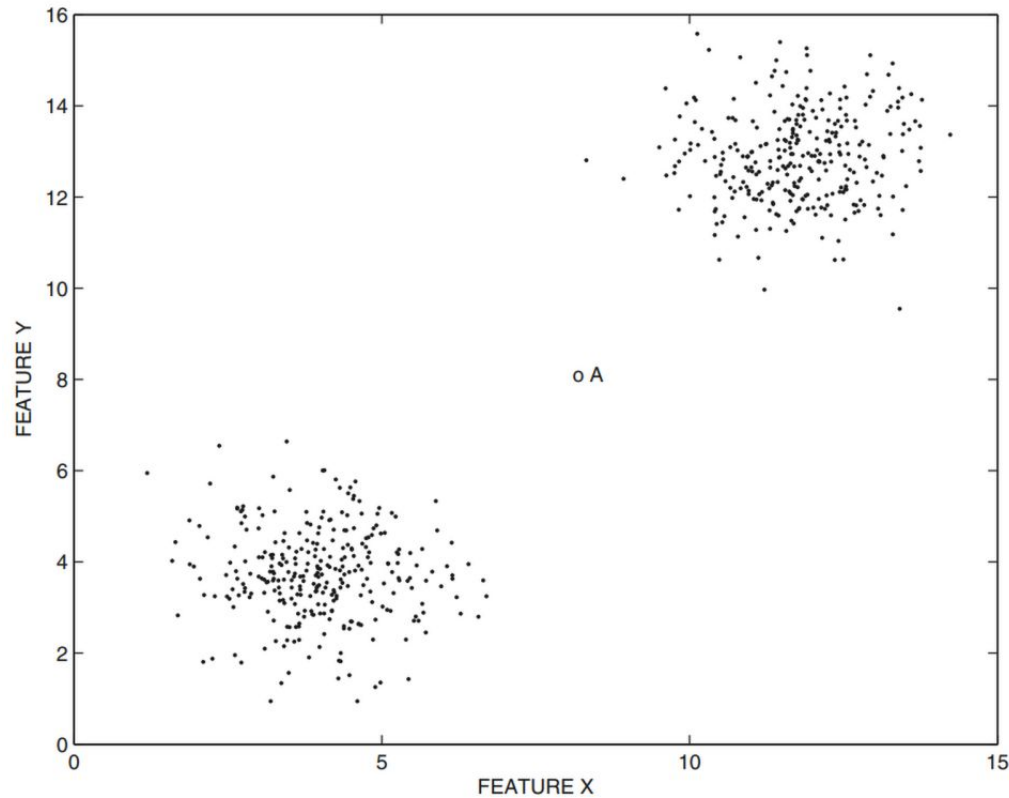


Abnormal Temperature

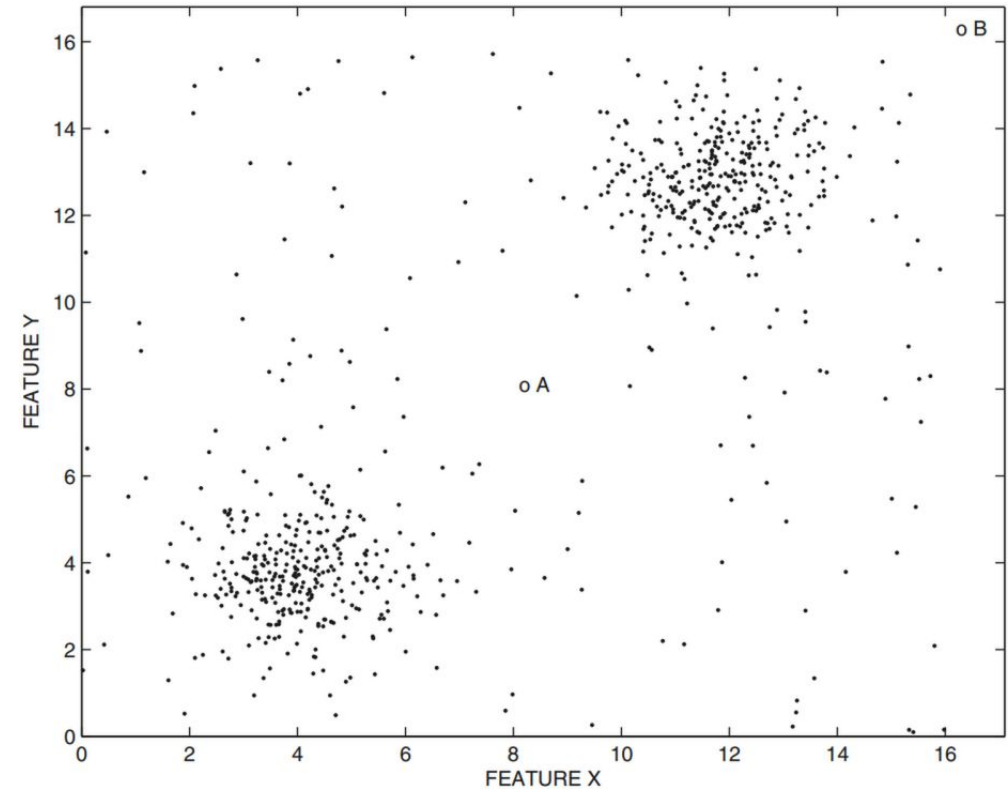
Proximity of Outliers



Noise and Anomaly

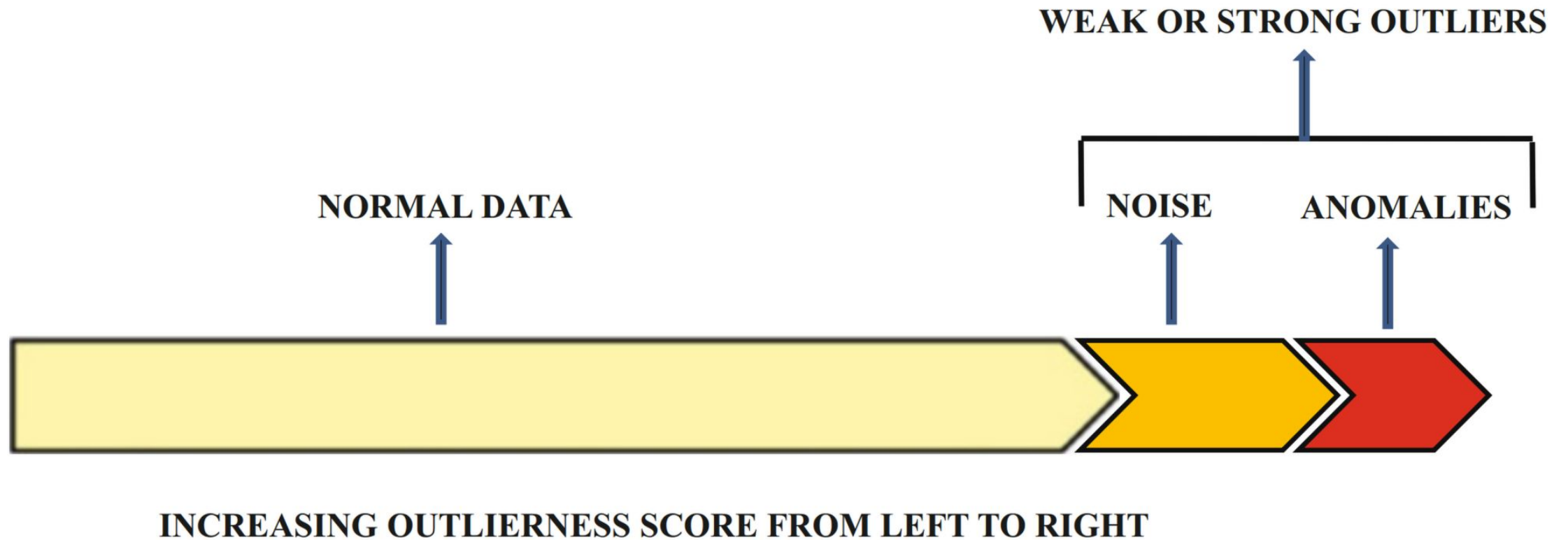


(a) No noise

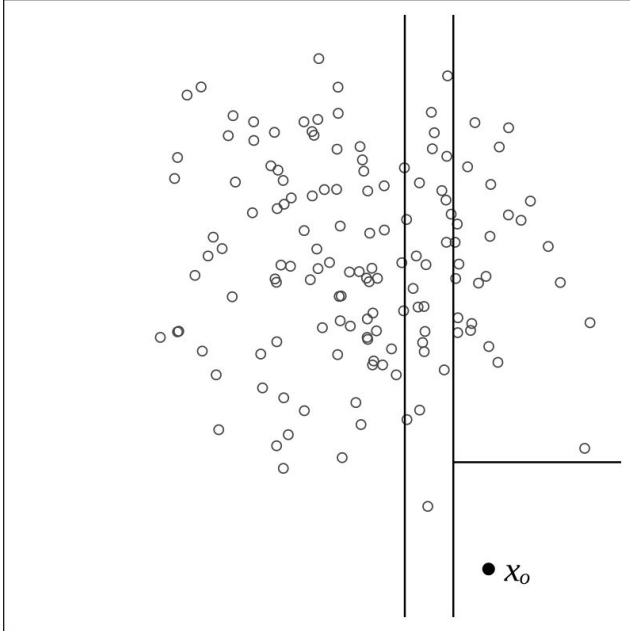


(b) With noise

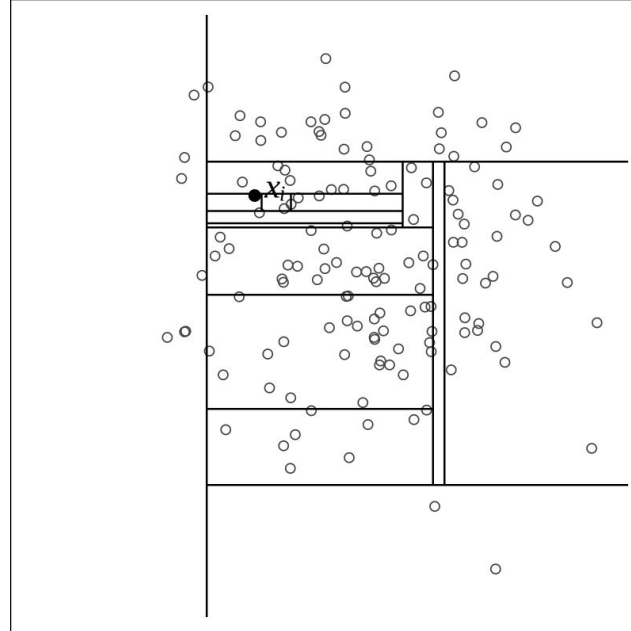
Outliers and Anomalies



Isolation Forest



Anomaly x_0 : isolated after a few partitions



Normal x_i : needs many more partitions

Build

- Ensemble of random binary trees on sub-samples
- Each node: random attribute, random split between min and max
- Grow until points are isolated or a height limit is reached

Score

- Average path length $h(x)$ over the forest
- Short path = high anomaly score, normalised to $[0, 1]$

Why it is the right foundation for streams

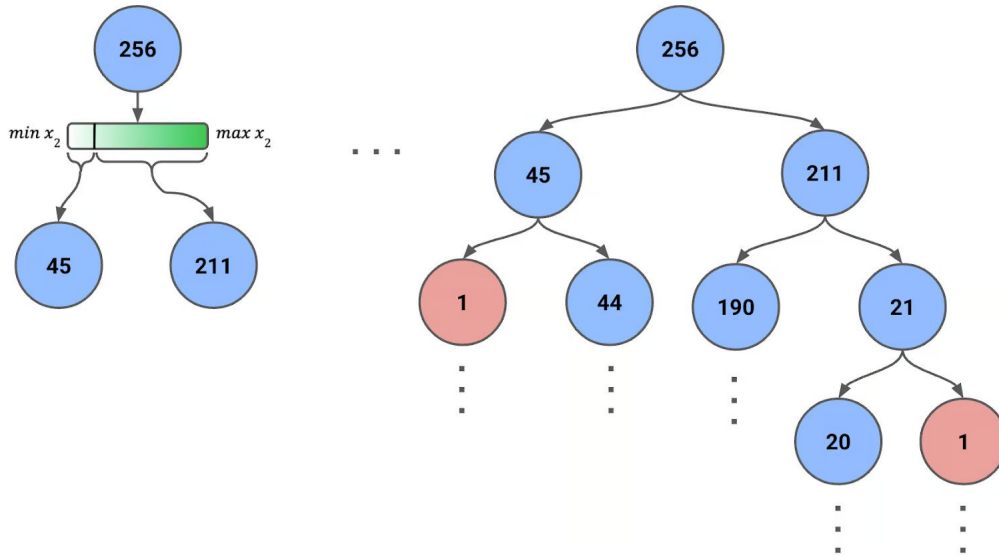
Linear time, no labels, no distance metric, small sub-samples suffice. The open question is how to keep the trees valid when the data cannot be stored and keeps changing.

Isolation Forest

Overview

Isolation Forest (iForest) is an ensemble method consisting of a collection of Isolation Trees.

An Isolation Tree (iTree) is a binary tree constructed based on sub-samples.



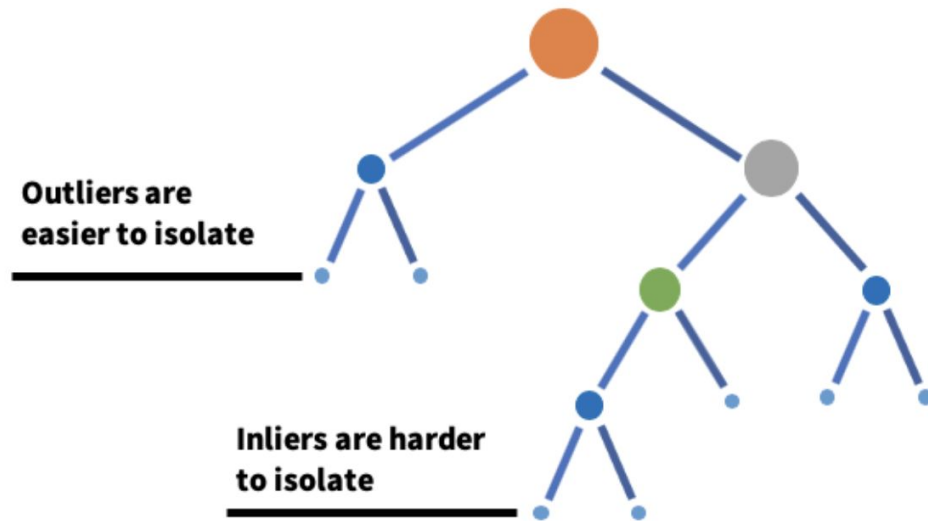
Train — To build an iTree

- At each node, randomly picks an attribute to split.
- Split point is uniformly (randomly) selected between the max and min values of the selected attribute.
- Grow the tree until all points are isolated (hence, Isolation Forest).

Evaluate — For each data point

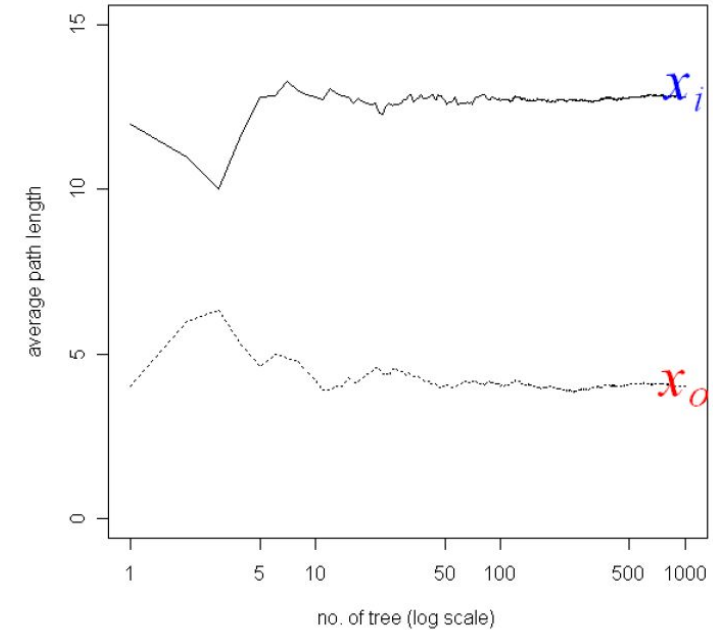
- Work out the path length from each iTree in the ensemble.
- Use the average path length across all trees to rank the anomaly score of the data point.

Isolation Forest



Normal points need more partitions to be isolated,
Anomalies require fewer.

Definition: Anomalies are **few** and **different**



(c) Average path lengths converge

Path length(No of partitions) converges with increasing
number of trees.

Data streams

1

Single pass, bounded resources

Each instance is seen once and cannot be revisited. Constant time and memory per instance, whatever the stream length.

Rules out any algorithm that needs a second pass or the full dataset in memory.

2

Concept drift

The distribution that generated yesterday's data is not the one generating today's. A model fixed at training time goes stale.

Forces a choice: what to keep, what to discard, and on what evidence.

3

Label scarcity

Ground truth arrives late, costs expert time, or never arrives at all. Any supervision has to fit a very small budget.

Makes the choice of which instances to label part of the algorithm.

4

Non-uniform anomalies

Anomalies are not spread evenly along the stream; they arrive in bursts, and a burst can dominate a whole window.

Breaks the assumption that any window is a representative sample.

Data streams

What a data stream is

A continuous, potentially unbounded sequence of instances that arrive over time and must be processed as they arrive, rather than a finite dataset available in full before training begins.

Where they come from

IoT and sensor networks	readings from thousands of devices, continuously
Financial transactions	payments authorised in milliseconds, around the clock
Network traffic monitoring	packets and flows at line rate, never replayed

Batch learning vs stream learning

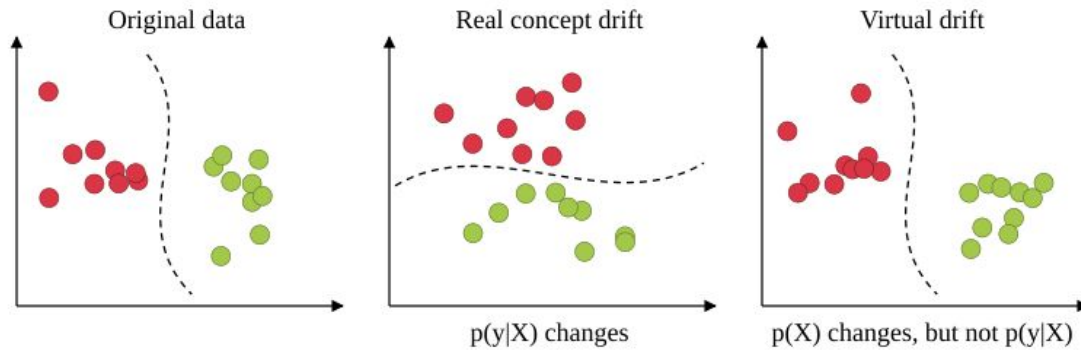
	Batch	Stream
Data	Finite, available in full	Unbounded, arrives over time
Passes	Many, over epochs	One, then the instance is gone
Memory	Grows with the dataset	Bounded, independent of length
Model	Fixed once trained	Updated as data arrives
Prediction	After training completes	At any moment

One pass over an unbounded sequence



Concept drift

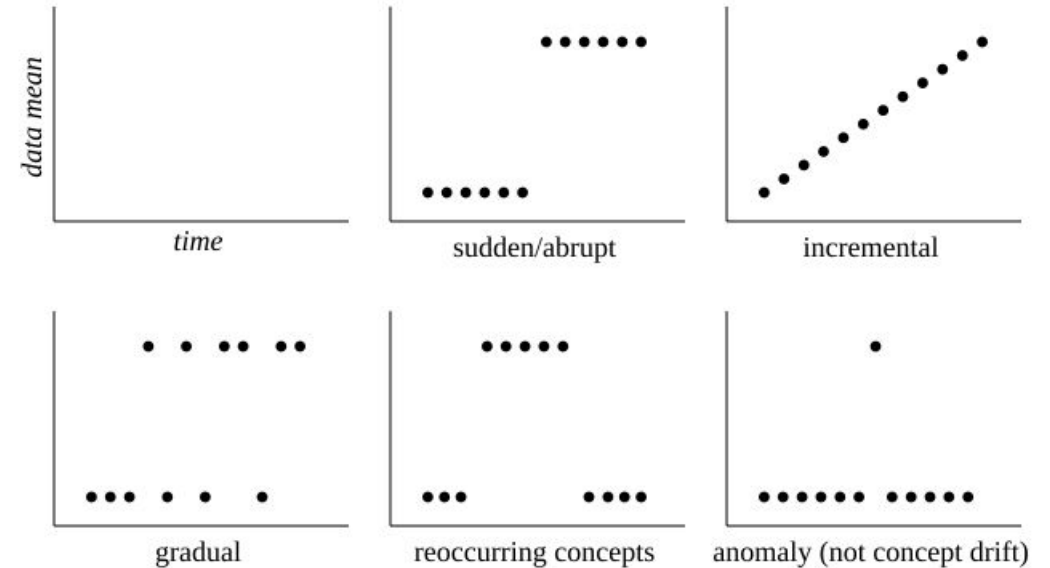
What changes



Real drift moves the boundary between normal and anomalous, so the existing model is wrong. Virtual drift changes the distribution of instances without moving that boundary.

Unsupervised detection sees only $p(X)$, so it cannot tell the two apart. Both must be handled by the same mechanism.

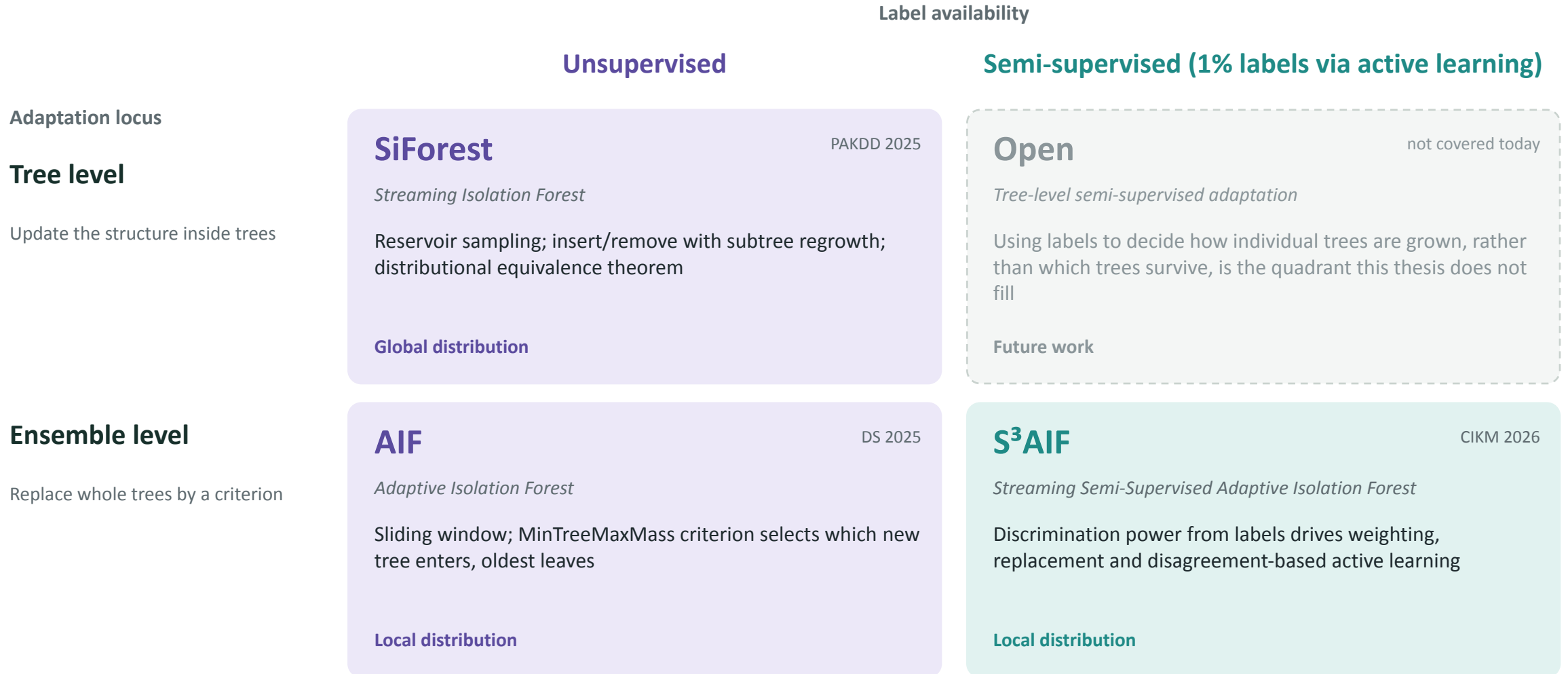
How it arrives



- Sudden** an abrupt switch from one concept to another
- Incremental** a transition through intermediate concepts
- Gradual** the new concept becomes more probable over time
- Recurring** a previous concept returns after an interval

An isolated outlier is an anomaly, not drift. A detector has to react to the second without over-reacting to the first.

Three algorithms on two design axes



All three share the isolation principle, single-pass processing, bounded memory, and an implementation in CapyMOA.

Part 1 of 2

Unsupervised streaming detection

SiForest: bring Isolation Forest to streams with a guarantee. AIF: decide which trees deserve to stay.

The unsupervised setting

Problem definition

Stream	$S = \{x_1, x_2, \dots, x_t, \dots\}$, $x_t \in \mathbb{R}^d$ arriving at time t
Windows	$S = \{W_1, W_2, \dots\}$, each of w consecutive instances: a shared timescale for batched updates
Hidden labels	$y_t \in \{0, 1\}$; $S = S_+ \cup S_-$ with $ S_- \ll S_+ $. No y_t is ever observed
Task	Build $\alpha: \mathbb{R}^d \rightarrow \mathbb{R}$ with $\alpha(x \in S_-) > \alpha(x \in S_+)$, so a threshold separates the two

A ranking objective, not a classification one: no threshold is committed to, which is why ROC-AUC and PR-AUC are the metrics.

Five requirements on α

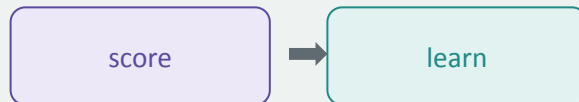
- **Single pass** each instance seen once
- **Low latency** processing time per instance minimal
- **Bounded memory** independent of stream length
- **Anytime availability** a valid score at any point
- **Adaptability** respond to concept drift

Together these rule out batch learning: no second pass, no full dataset, no offline training phase.

Evaluation

1 Prequential evaluation

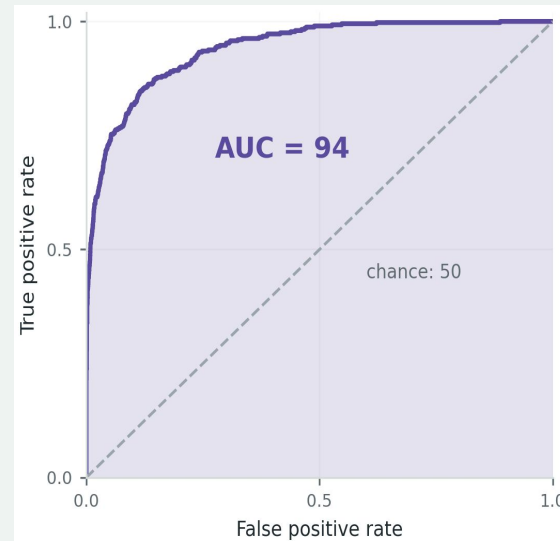
Every instance is used for testing first, to produce a score, and only then for training. It respects the temporal order and needs no held-out split, which matches how the model would actually be deployed.



Nothing is held back for a test set, and every score is produced by a model that has not yet seen that instance. Over a long stream that is a very large number of honest predictions.

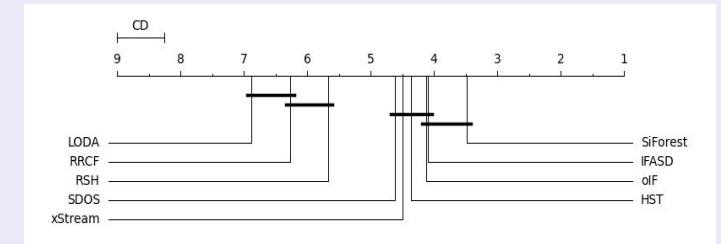
2 ROC-AUC

The probability that a randomly chosen anomaly is scored above a randomly chosen normal instance.



50 is chance, 100 is a perfect ranking.

3 Reading a critical difference diagram



Four of the results slides use one of these.

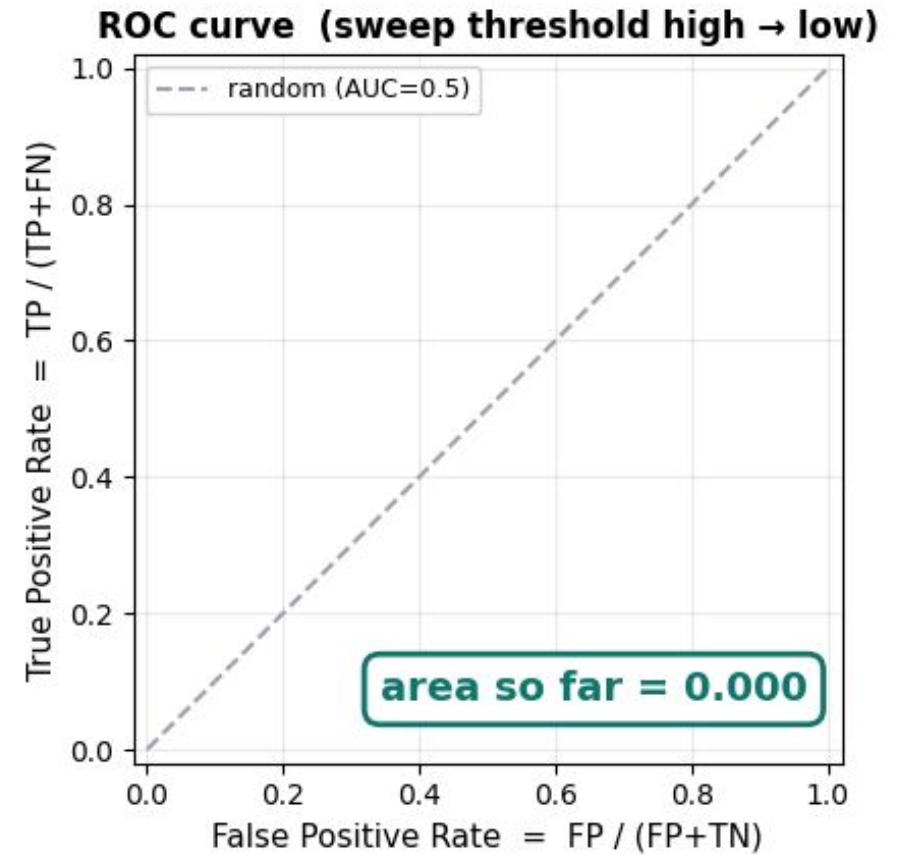
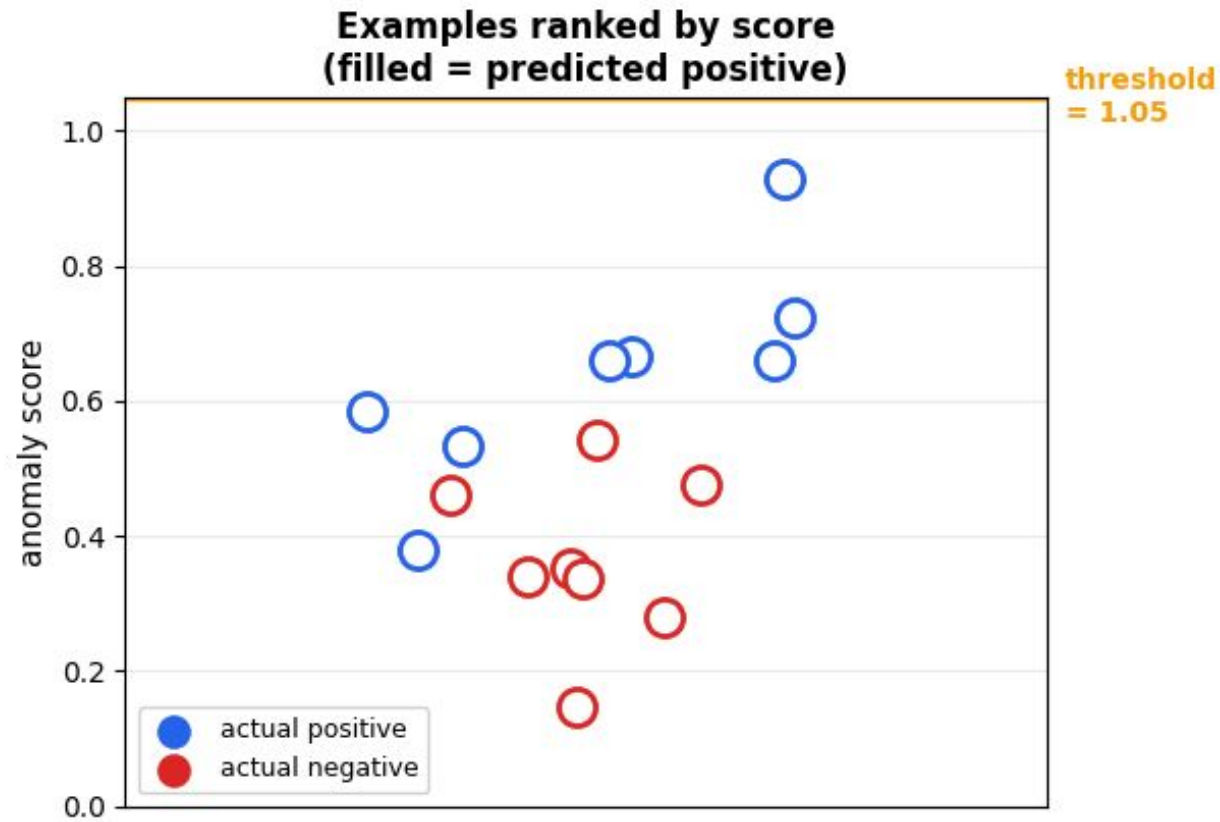
Position average rank over all datasets; further right is better

Bar methods joined by a bar are not significantly different

CD the gap needed for significance, from Friedman then Nemenyi at $\alpha = 0.05$

ROC-AUC

Drawing the ROC curve & computing ROC-AUC



Benchmark datasets

Dataset	#Inst.	#Feat.	%Anom.
ALOI	49,534	27	3.02
backdoor	95,329	196	2.44
campaign	41,188	62	11.27
celeba	202,599	39	2.24
census	299,285	500	6.20
cover	286,048	10	0.96
donors	619,326	10	5.93
fraud	284,807	29	0.17
Friday	191,033	78	1.03
http	567,498	3	0.39
insects	212,514	33	5.51
magic	19,020	10	35.16

Dataset	#Inst.	#Feat.	%Anom.
mammo	11,183	6	2.32
mulcross	262,144	4	10.00
NYC	10,320	1	5.19
NYC-S	10,273	48	5.22
shuttle	49,097	9	7.15
skin	245,057	3	20.75
smtp	95,156	3	0.03
Thu-Inf	288,602	78	0.01
Thu-WA	170,366	78	1.28
Tue	445,909	78	3.10
Wed	692,703	78	36.48

The unsupervised streaming baselines

Isolation-based

- HST** Random split at the midpoint of each attribute's range; scores by node mass rather than path length. Needs feature ranges up front.
- IFASD** Batch iForest over a sliding window, rebuilt in full whenever the anomaly rate signals drift.
- RRCF** Maintains trees incrementally, but picks the split attribute in proportion to its range.
- oIF** Dynamic multi-resolution histograms per node, updated incrementally over a sliding window.

Projection and hashing

- LODA** An ensemble of sparse random linear projections, each with a one-dimensional histogram; scores are mean log-likelihoods.
- xStream** Sparse projections into low-dimensional subspaces, then half-space chains for density. Constant time and space per instance.
- RSH** Random subspaces discretised by hash functions into histograms that approximate local density.

Distance-based

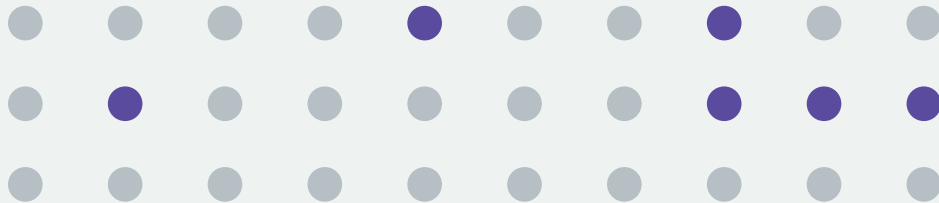
- SDOS** Keeps a set of observers summarising the distribution; scores by distance to the nearest observer, with an ageing mechanism.
- iMF** Isolation with Mondrian processes: partitions stay consistent across sample sizes, but the cost is high. Reported per dataset and excluded from the averages and rankings, because it often failed to finish within the time budget.

Getting a random sample: batch vs stream

Unsupervised · tree level

Batch: uniform sampling

The dataset is finite and sits in memory. Pick k of the n instances at random.



all n instances available at once; k chosen at random

Why it works

n is known, so every instance can be given the same chance of selection. Isolation Forest assumes exactly this kind of sample.

Stream: reservoir sampling

Instances arrive one at a time and n is never known. Keep a reservoir of k , and update it as they pass.

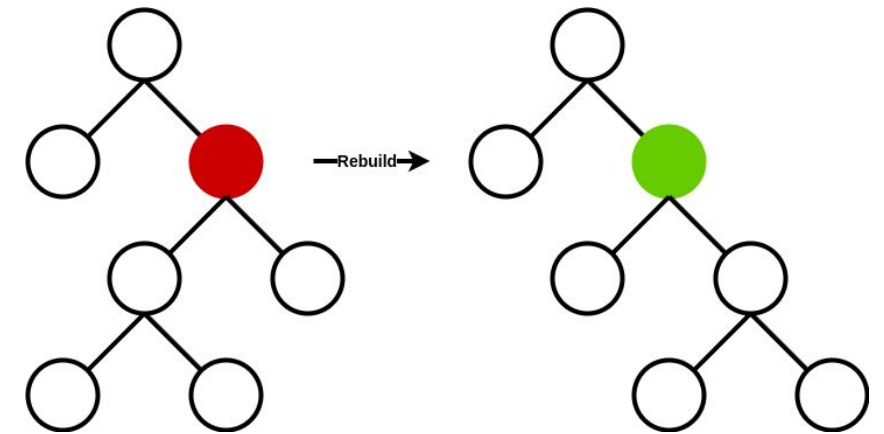
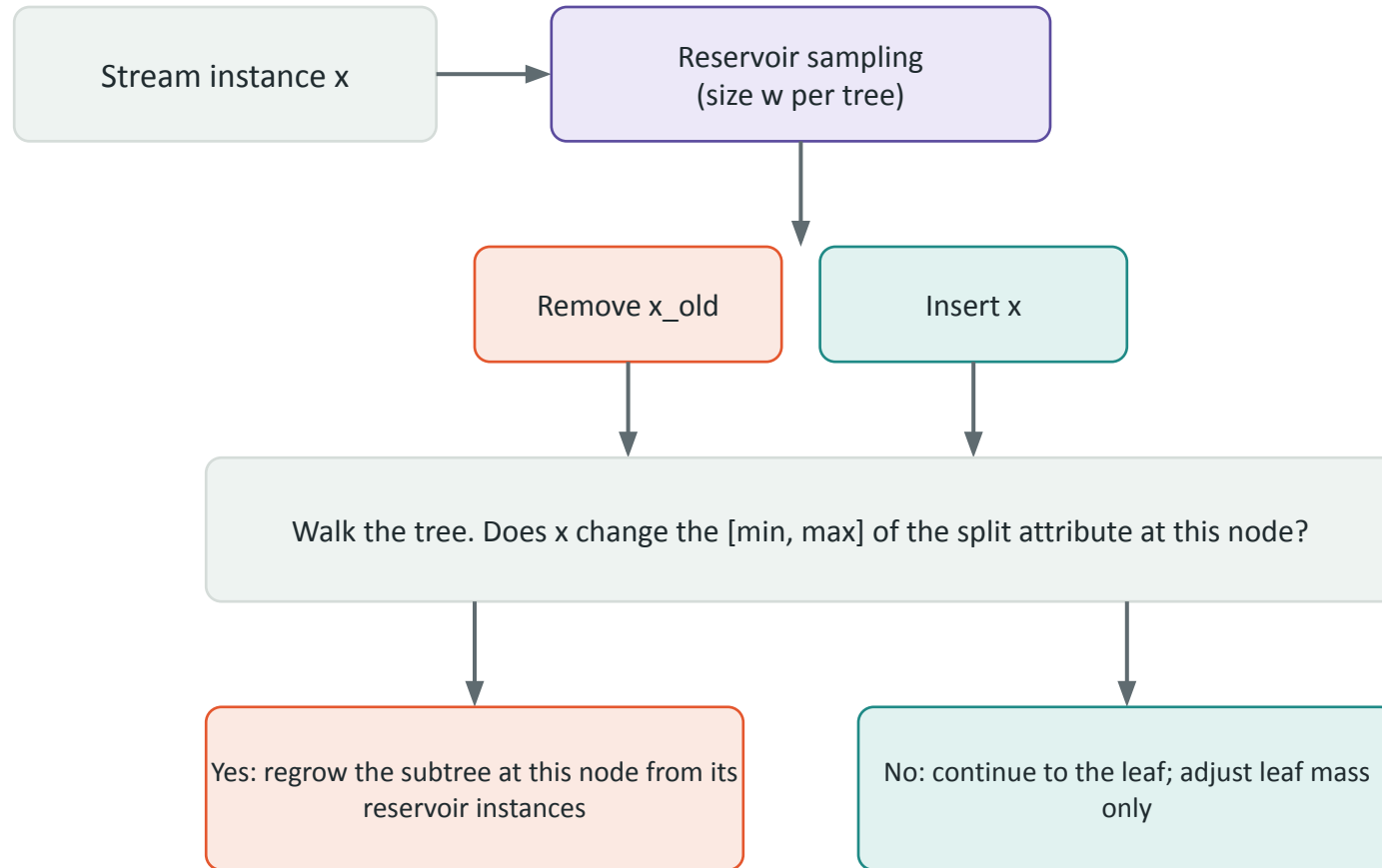


A kept instance replaces one already in the reservoir. Nothing else is stored, ever.

Why it works

At every point, the k instances held are a uniform random sample of everything seen so far, without ever knowing how much that will be.

SiForest: reservoir sampling with dynamic tree maintenance



SiForest: a guarantee for incremental maintenance

Distributional equivalence theorem

For a SiTree T built on reservoir D :

Removing x by Algorithm Remove yields T' with $P(T') = P(T(D \setminus \{x\}))$

Inserting x by Algorithm Insert yields T'' with $P(T'') = P(T(D \cup \{x\}))$

Why it holds

- Reservoir sampling keeps a uniform random sample of everything seen so far, which is what a batch iTree would be built on
- An update changes a node only when it changes the $[\min, \max]$ range of that node's split attribute; the affected subtree is then regrown from the reservoir instances that fall in it
- Every other node keeps a split that is still uniform over an unchanged range, so its distribution is untouched

What it buys

No loss from going incremental

The maintained ensemble is not an approximation of the batch model; it has the same distribution

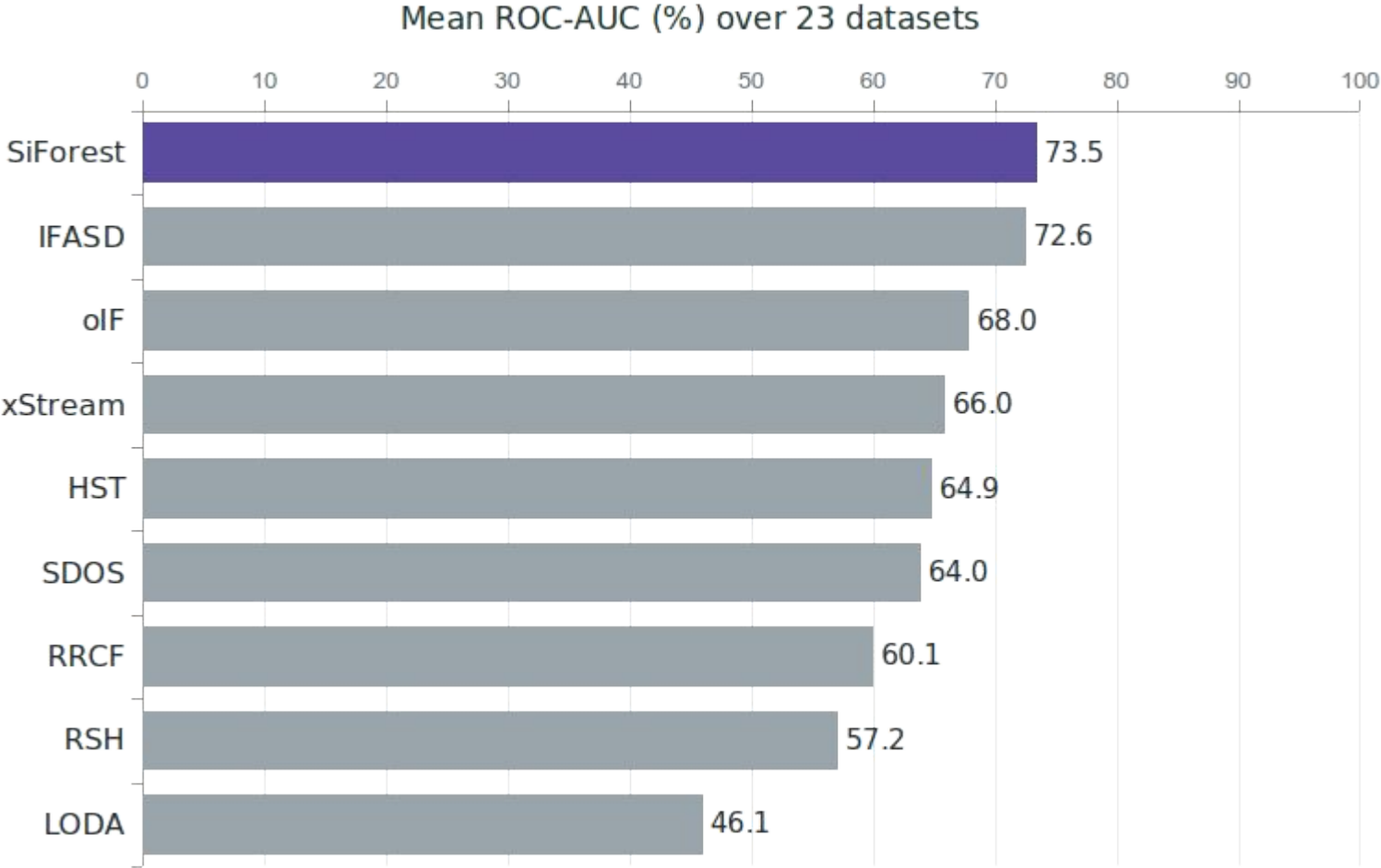
No storage of the stream

Only T reservoirs of w instances, independent of how long the stream runs

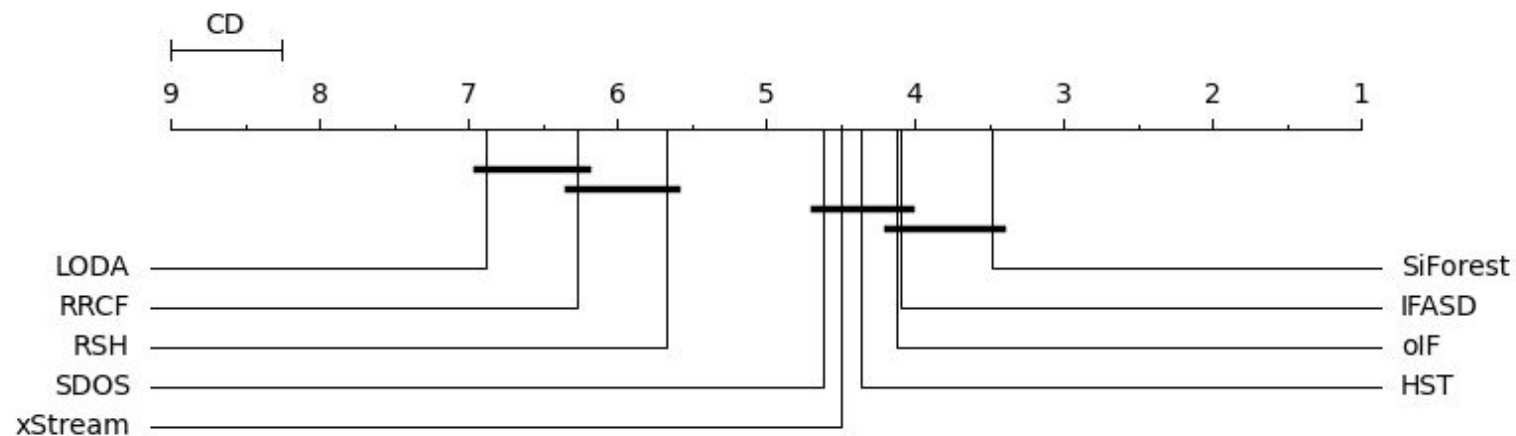
No assumption on arrival order

The result holds for any sequence of insertions and removals

SiForest: mean ROC-AUC against every baseline

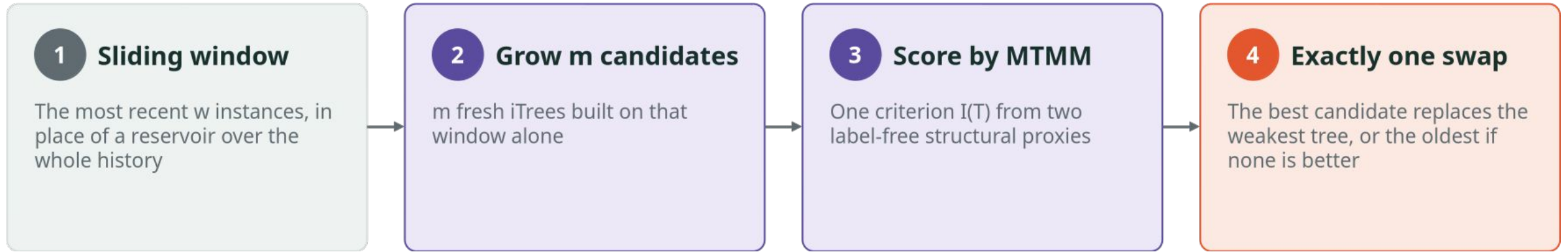


SiForest: best mean ROC-AUC of nine streaming detectors



Critical difference diagram, ROC-AUC ranks. Methods joined by a bar are not significantly different; lower rank is better.

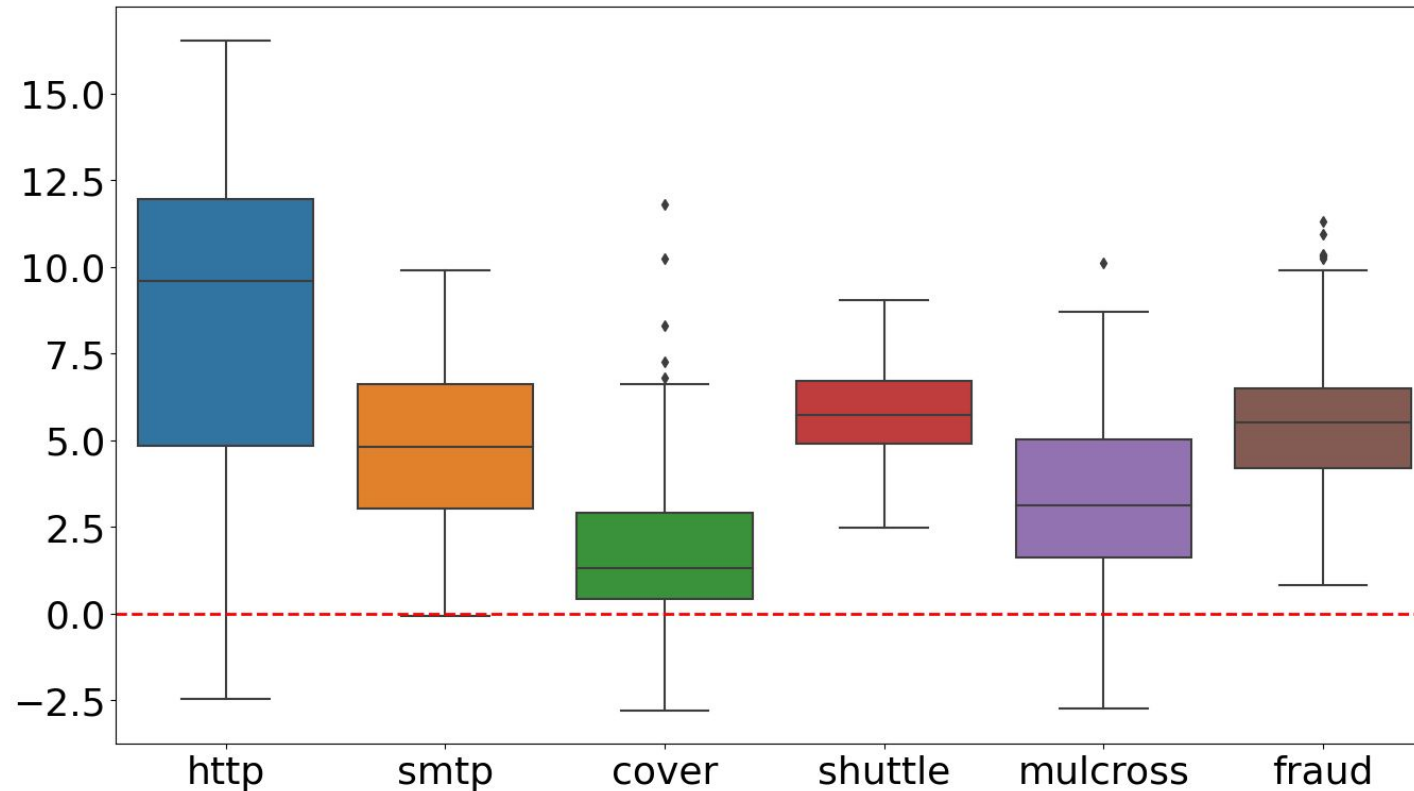
Adaptive Isolation Forest



Incremental Ensemble Updating Mechanism

- **Adaptive Training:** New trees are trained exclusively on current data windows to ensure the model aligns with the latest data distribution.
- **Ensemble Optimization:** High-scoring trees are preserved to enhance overall ensemble performance.
- **Concept Evolution:** Older trees are systematically retired to discard outdated concepts.

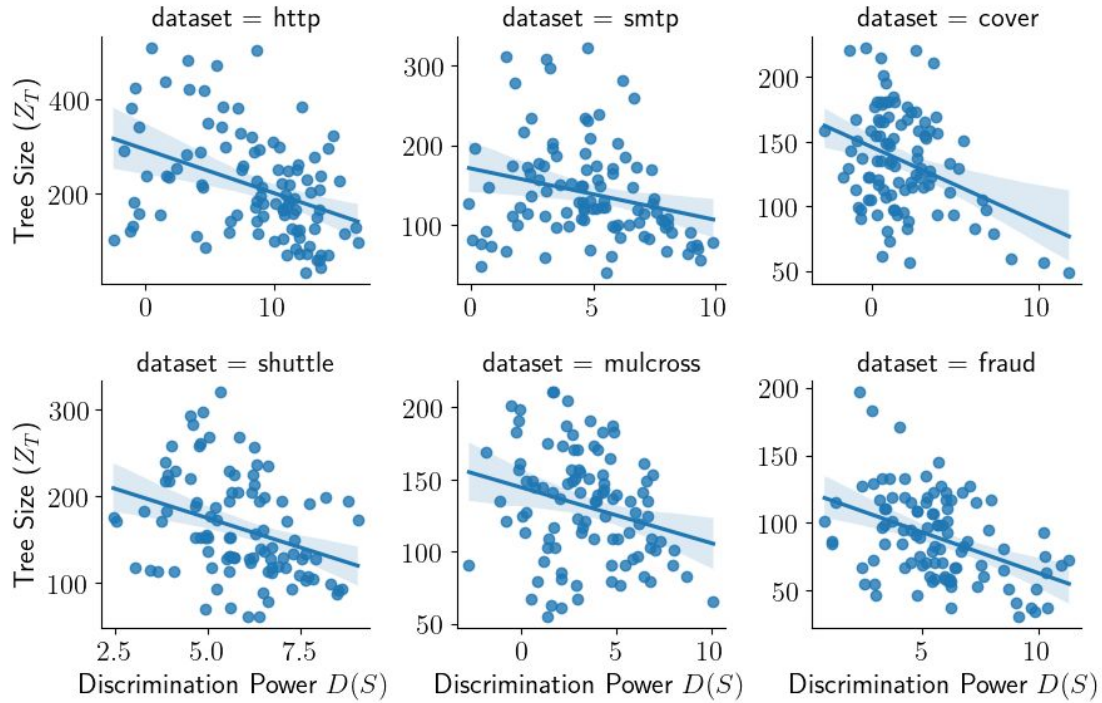
Trees in one forest differ widely in discrimination power



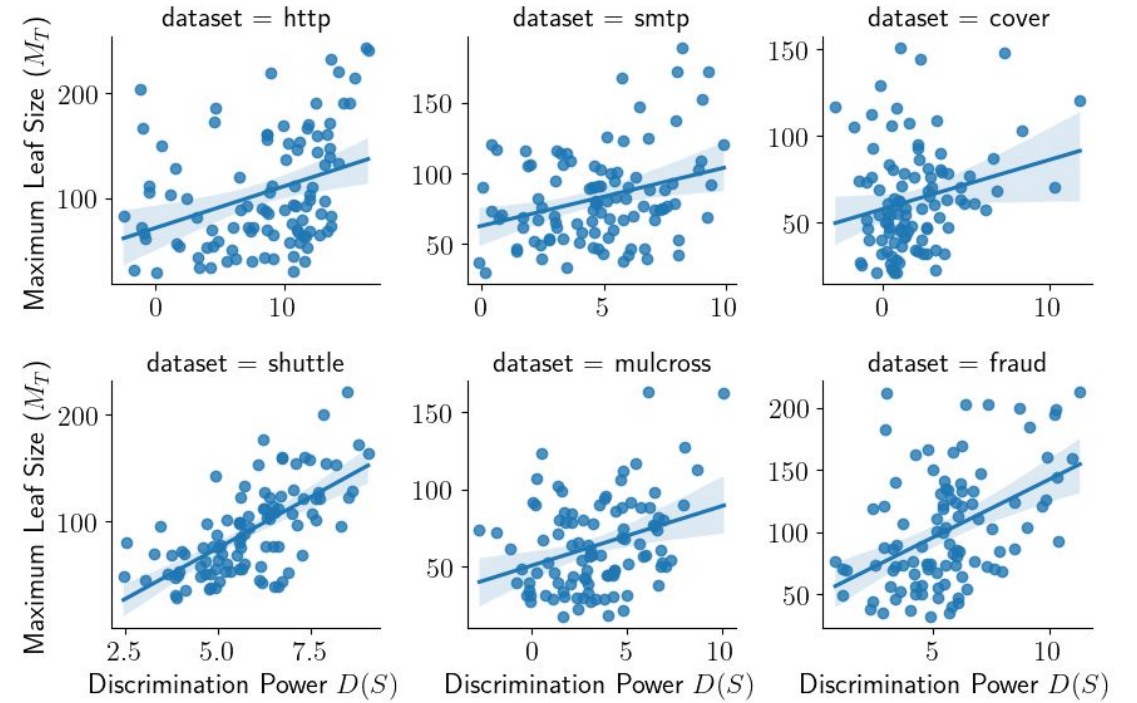
Discrimination power $D_T(S)$ of 100 iTrees per dataset. Above the red line the tree separates anomalies correctly; below it, the tree ranks anomalies as more normal than normal instances.

$$D_T(S) = \frac{1}{|S_n|} \sum_{\mathbf{x}_n \in S_n} PL_T(\mathbf{x}_n) - \frac{1}{|S_a|} \sum_{\mathbf{x}_a \in S_a} PL_T(\mathbf{x}_a),$$

MTMM: two label-free proxies for discrimination power



Tree size Z_T : negatively correlated. Small trees isolate anomalies early.

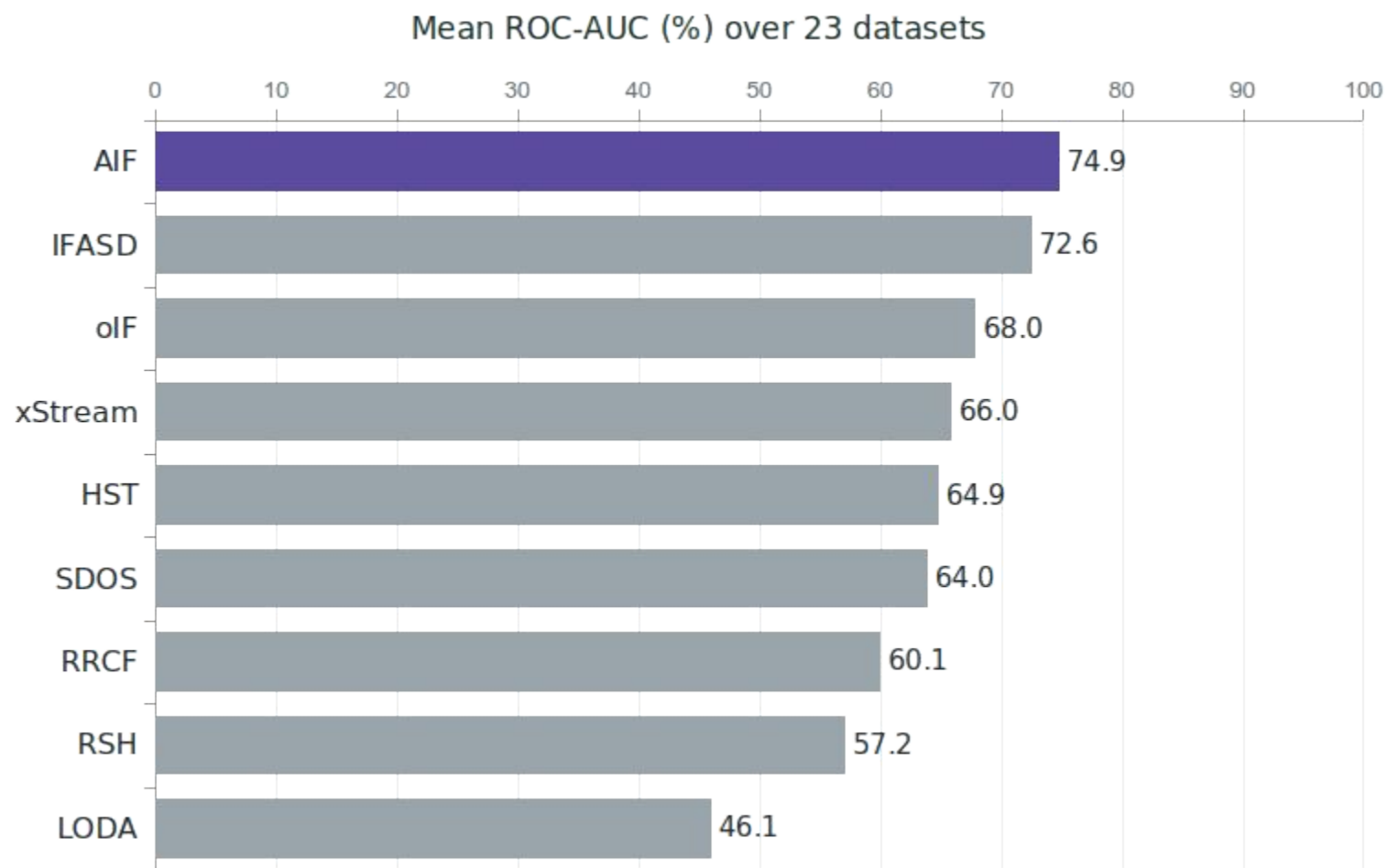


Max leaf mass M_T : positively correlated. Dense normal regions stay in one leaf.

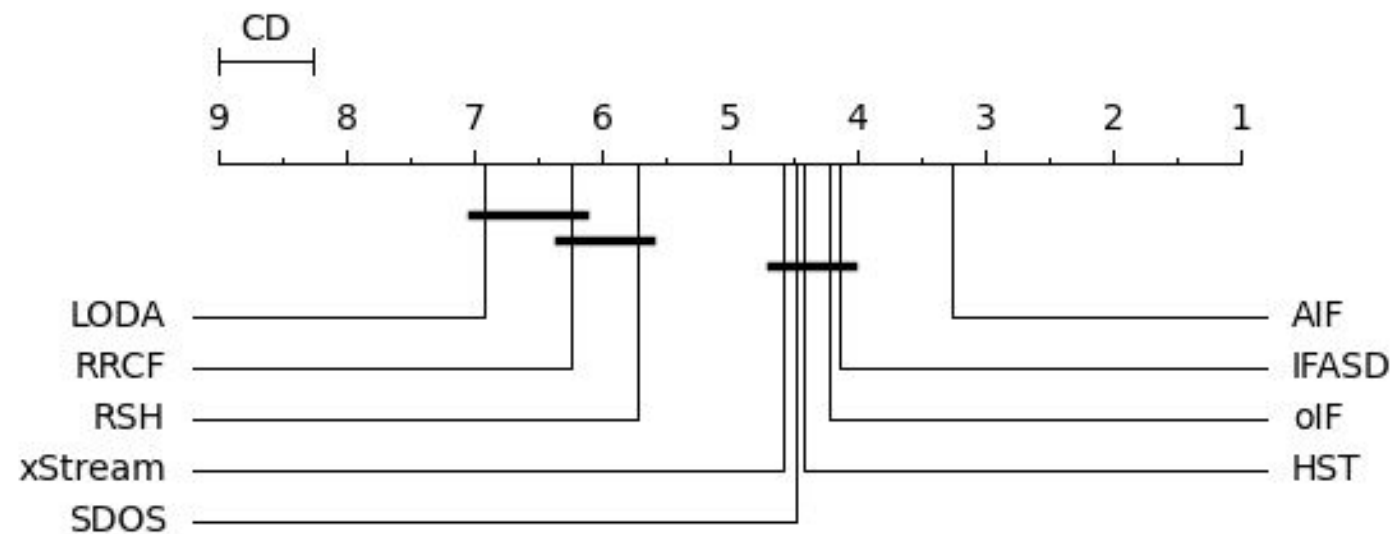
$$I_{\beta}(T) = \beta (1 - Z_T) + (1 - \beta) M_T$$

Tree size and maximum leaf mass are min-max normalised over the m candidates. The candidate with the highest I enters the ensemble.

AIF: mean ROC-AUC against every baseline



AIF: significantly ahead of eight baselines



AIF ranks first and its gap to every baseline exceeds the critical difference.

Part 2 of 2

Semi-supervised streaming detection

Spend a 1% labelling budget where it matters, then let the labels drive scoring, replacement, and construction.

The semi-supervised active learning setting

Carried over from the unsupervised setting

Stream	$S = \{x_1, x_2, \dots\}, x_i \in \mathbb{R}^d$
Windows	$S = \{W_1, W_2, \dots\}$, each of w instances
Hidden labels	$y_i \in \{0, 1\}; S_a \ll S_n $
Task	$\alpha: \mathbb{R}^d \rightarrow \mathbb{R}$ ranking anomalies above normal instances
Requirements	Single pass, low latency, bounded memory, anytime availability, adaptability

Added for semi-supervised active learning

Oracle	$\mathcal{O}$ returns the true label $y_i = \mathcal{O}(x_i)$ of any queried instance, promptly
Query strategy	In window W_i , strategy π selects $Q_i \subset U_i$ of size b_i from the unlabelled instances U_i
Label buffer	Answers accumulate in a sliding buffer $\mathcal{L}$; unqueried instances are discarded when the window closes
Global budget	$\sum_i b_i \leq B = \rho \cdot S $, with $\rho = 1\%$ throughout this work
Extended objective: rank anomalies above normal instances, use the labels in $\mathcal{L}$ effectively, and maximise the information gained per query within B .	

S³AIF: Discrimination power as the organizing principle

Discrimination Power

$$D_T = \text{mean } h_T(x) | x \in L_n - \text{mean } h_T(x) | x \in L_a$$

- $h_T(x)$: path length of sample x in tree T .
- L_n, L_a : sliding buffers of labelled normal instances and anomalies.
- Large D_T indicates the tree isolates anomalies much earlier than normals.
- S³AIF computes this directly, whereas AIF could only approximate it with MTMM.

1 Strategic replacement

At every window, replace:

$$\text{argmin}_T D_T$$

with a fresh tree built on the current window.

*Quality, not age, decides who leaves.
Ensemble size T remains unchanged.*

2 Weighted scoring

$$\text{Score}(x) = 2 - (d \cdot h_x) / (\sum d \cdot c(w))$$

Each tree contributes in proportion to its discrimination power.

$$d_T = \max(D_T, 0)$$

This ensures that ineffective trees are completely silenced.

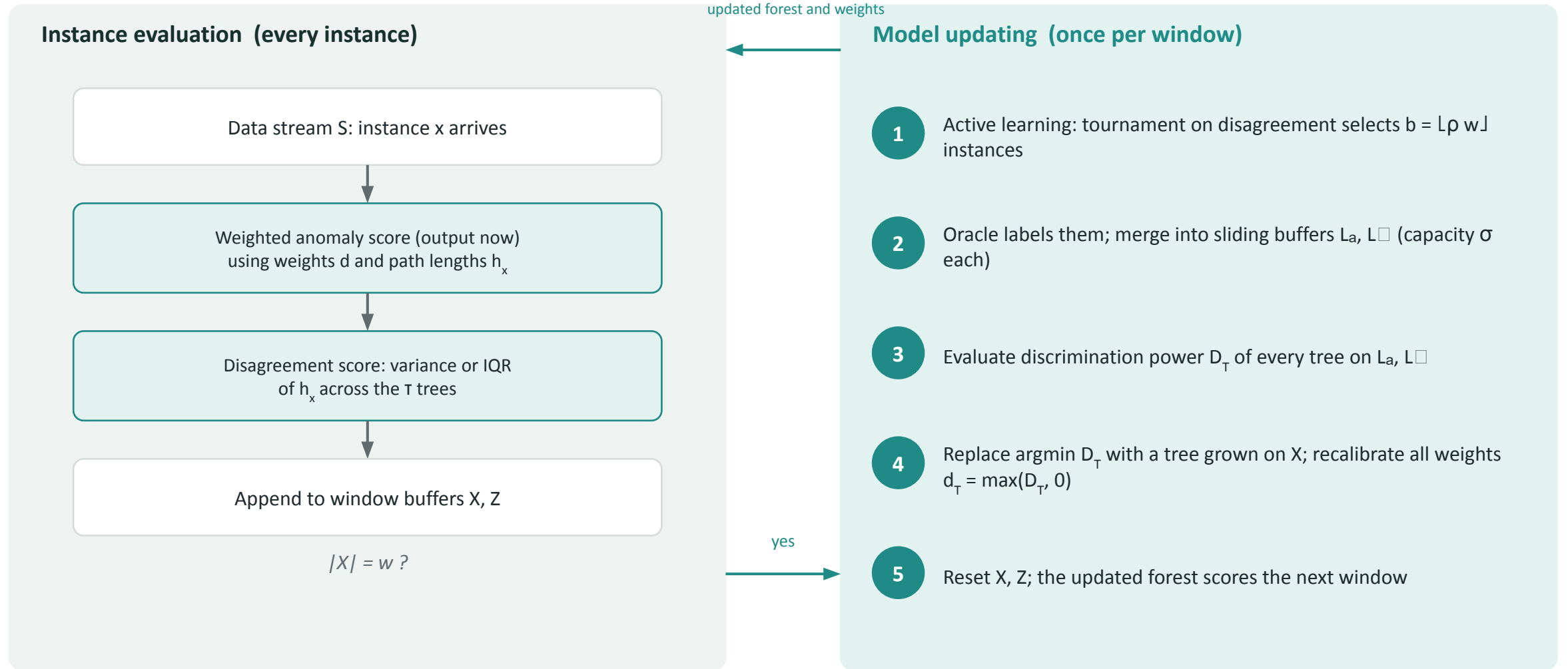
3 Aligned active learning

Targeted Label Querying:

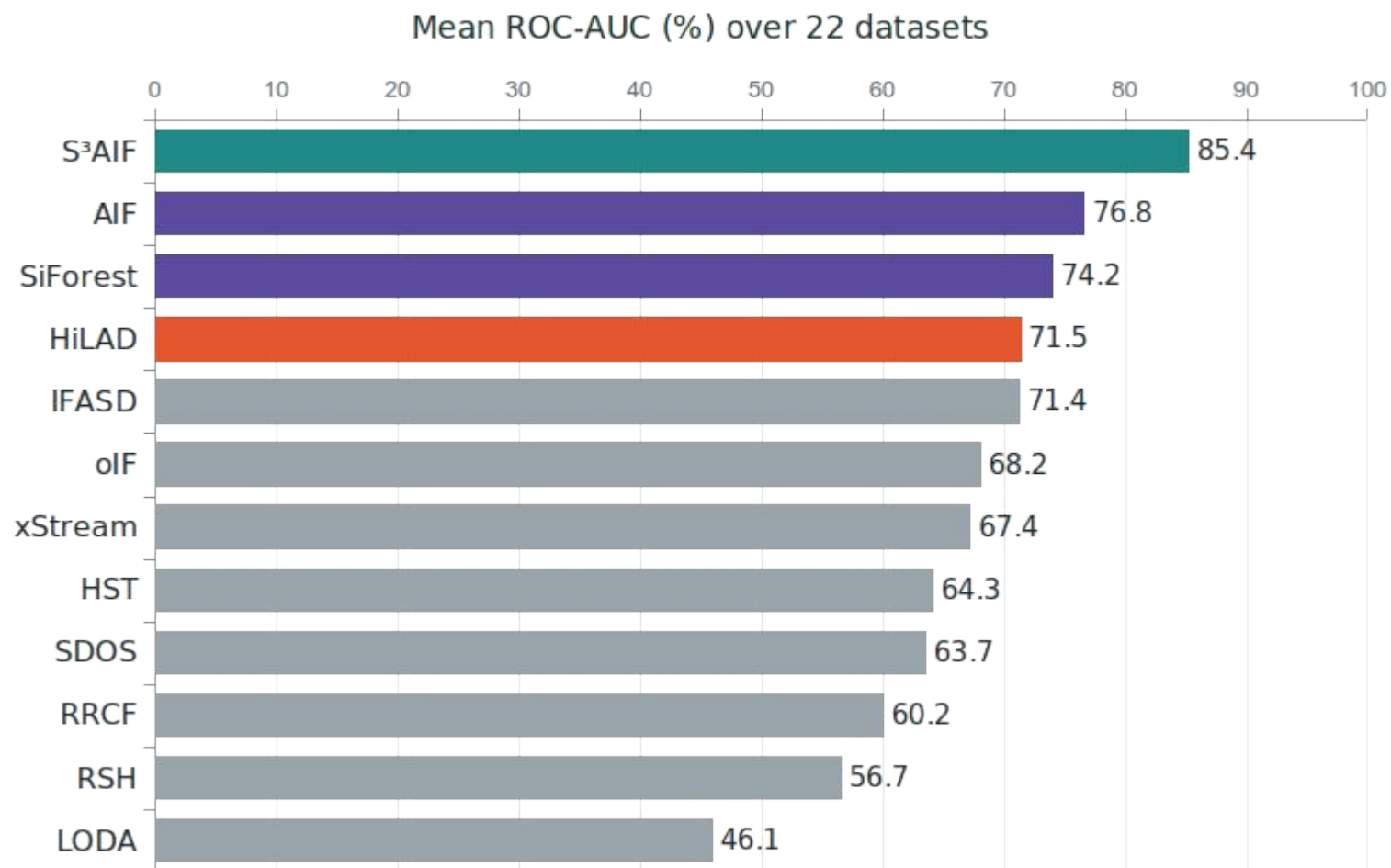
Query labels where trees disagree on path length (evaluated via variance or IQR across the ensemble).

Disagreement serves as the same underlying signal that discrimination power rewards.

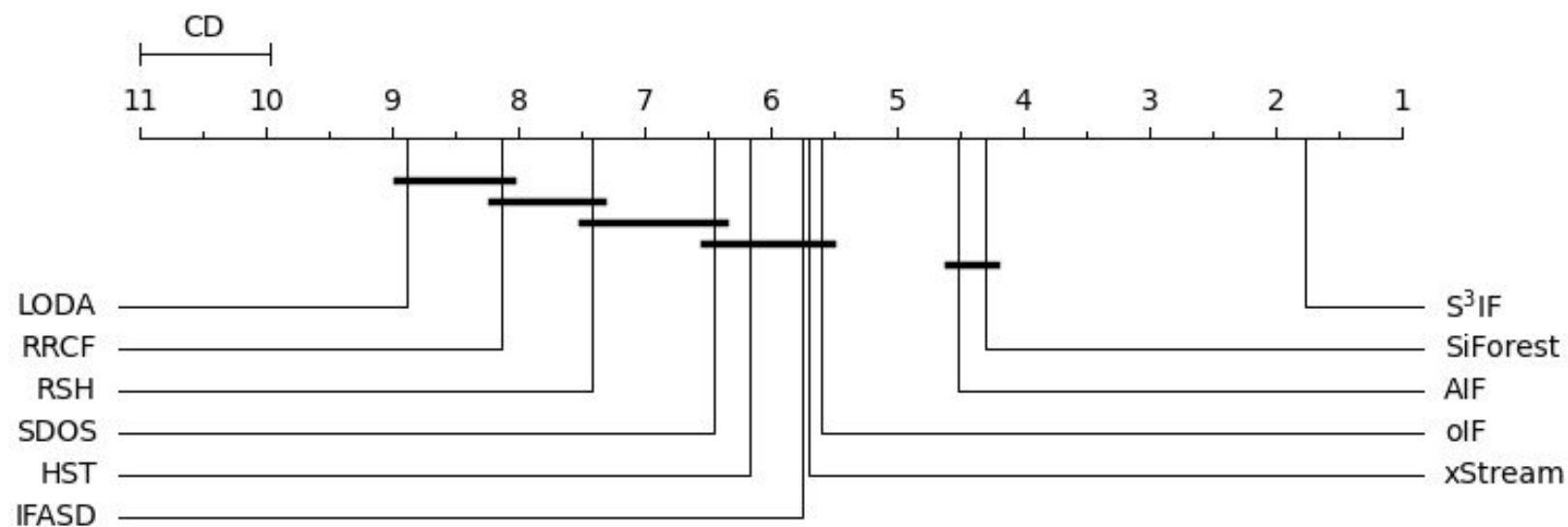
S^3 AIF: two phases, one feedback loop



S^3 AIF: mean ROC-AUC across the field



S^3 AIF: beats HiLAD and all unsupervised baselines



ROC-AUC ranks vs unsupervised streaming baselines.

Every gap exceeds the critical difference of 1.018

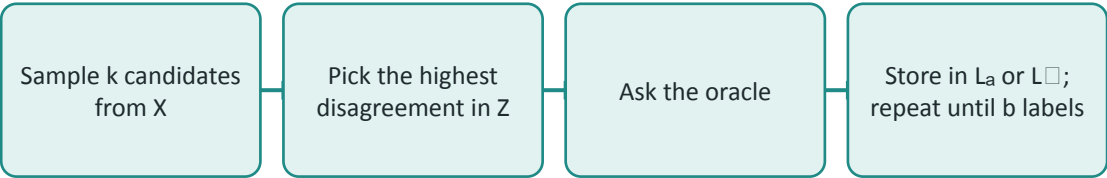
PR-AUC gives the same ordering.

Tournament active learning on path-length disagreement

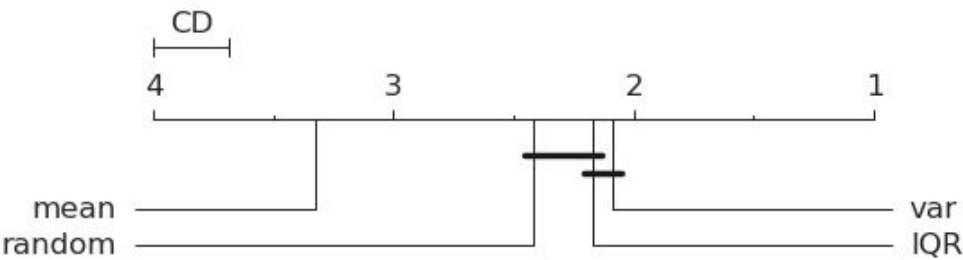
Why disagreement

- Global anomaly: short path in nearly every tree, low spread. Already confidently scored; a label adds little
- Local anomaly or boundary case: early in some trees, deep in others, high spread. A label here changes the model
- Var: sensitive to extreme trees. IQR: consensus of the central 50%, robust to unreliable trees

Tournament



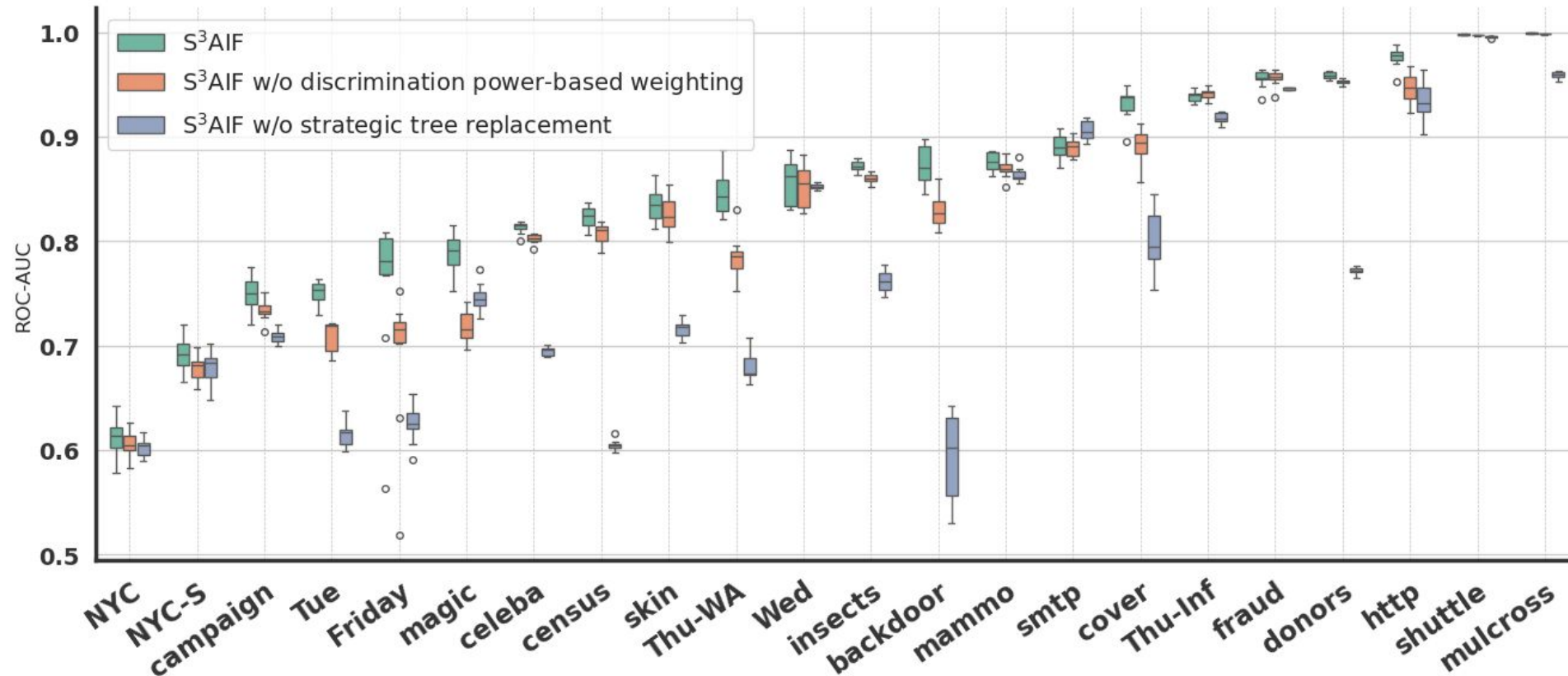
k random draws keep exploration; choosing the winner keeps exploitation. With $\rho = 1\%$ and $w = 256$: $b = 2$ queries per window, 0.78% of the stream labelled.



Strategy	Mean ROC-AUC	Mean rank
Var tournament (S^3 AIF)	85.4	2.1
IQR tournament	85.5	2.2
Random selection	85.3	2.4
Mean path length	83.4	3.3

Both tournaments beat mean-path selection; Var is significantly ahead of both baselines (CD = 0.316). Var is used in all remaining experiments.

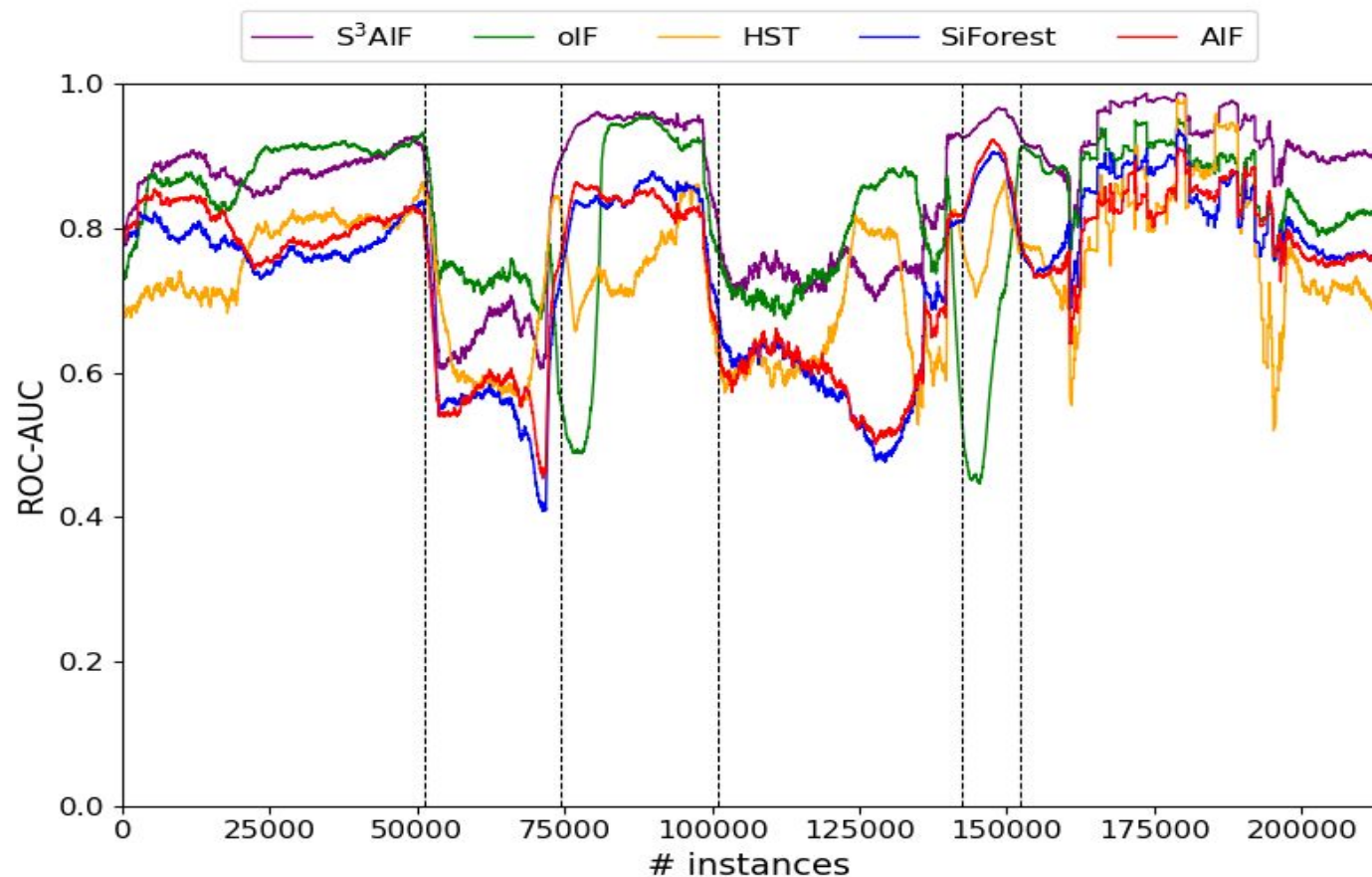
S^3 AIF: what each component contributes



ROC-AUC per dataset. Green: full S^3 AIF. Orange: no discrimination-power weighting. Blue: no strategic replacement (oldest tree replaced instead).

Both components help, but they are not equal: weighting refines the score, while strategic replacement is what keeps the ensemble healthy over time.

S³AIF under concept drift



ROC-AUC over the insects stream. Dotted lines mark distributional changes.

Everyone drops at a shift

No method here anticipates a change it has not seen; none holds its level immediately after one.

S³AIF recovers faster

It returns to a high level sooner and stays above the baselines in the stable stretches between drifts.

Three algorithms side by side

	SiForest	AIF	S ³ AIF
Supervision	Unsupervised	Unsupervised	Semi-supervised
Data view	Reservoir sampling	Sliding window	Sliding window
Distribution focus	Global (cumulative)	Local (recent)	Local (recent)
Adaptation mechanism	Cumulative subtree regrowth	MTMM criterion	Discrimination power
Adaptation level	Tree	Ensemble	Ensemble
Active learning	None	None	Disagreement-driven
Tree construction	Random splits	Random splits	Random splits
Anomaly scoring	Path length	Path length	Weighted by tree power
Amortised update	$O(w \log_2 w)$	$O(m \log_2 w)$	$O(\log_2 w)$
Space	$O(T w)$	$O((T + m) w)$	$O(T w + 2\sigma)$
Research question	RQ1	RQ2	RQ2, RQ3
Mean ROC-AUC	73.5% (23 datasets)	74.9% (23 datasets)	85.4% (22 datasets)

Limitations and future work

Limitations across the three methods

- Cold start: an anomaly burst in the first window contaminates the initial ensemble
- Fixed window size, regardless of the drift rate
- One tree replaced per adaptation step may be too slow under strong drift
- Point anomalies only; contextual and collective anomalies are out of scope
- Deployment: delayed or noisy labels, evolving feature space

Future work

Short term

- Recency-biased or importance sampling for SiForest, keeping the guarantee
- MTMM as a criterion for other streaming ensembles
- Cold-start screening or rapid relabelling of the first window
- Dynamic labelling budget: more queries during drift, fewer when stable

Longer term

- Feature-level adaptation: using labels to weight or select features during tree construction
- Adaptive thresholds to turn scores into decisions under drift
- Deep representations with the isolation principle for high-dimensional streams

Publications and open-source implementation

Published

Streaming Isolation Forest

PAKDD 2025, LNCS 15870. Liu, Cassales, Liu, Pfahringer, Bifet.

Published

Adaptive Isolation Forest

Discovery Science 2025, LNCS 16090. Liu, Cassales, Liu, Pfahringer, Bifet.

Accepted

Streaming Semi-Supervised Adaptive Isolation Forest

CIKM 2026. Liu, Cassales, Liu, Nowaczyk, Pfahringer, Bifet. doi:10.1145/3799682.3840949

CapyMOA

Open-source Python framework for streaming machine learning

- All three algorithms implemented in Python on CapyMOA
- SiForest and AIF already released; S³AIF to follow
- Same evaluation harness for every baseline in the thesis
- capymoa.org

Funding: TAI AO programme, New Zealand Ministry of Business, Innovation and Employment (MBIE).

Thank you

Questions?

Jia (Justin) Liu · University of Waikato · jia.liu@waikato.ac.nz · Code: capymoa.org