† POLITECNICO MILANO 1863
‡ THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Online Isolation Forest

Filippo Leveni †✉, Guilherme Weigert Cassales ‡, Bernard Pfahringer ‡, Albert Bifet ‡,
Giacomo Boracchi †

ICML International Conference On Machine Learning
41st International Conference on Machine Learning (ICML)
Vienna, Austria, *July* 21-27 2024

Online Anomaly Detection

Online Anomaly Detection

Online Anomaly Detection

Online Anomaly Detection

Online Anomaly Detection

Online Anomaly Detection

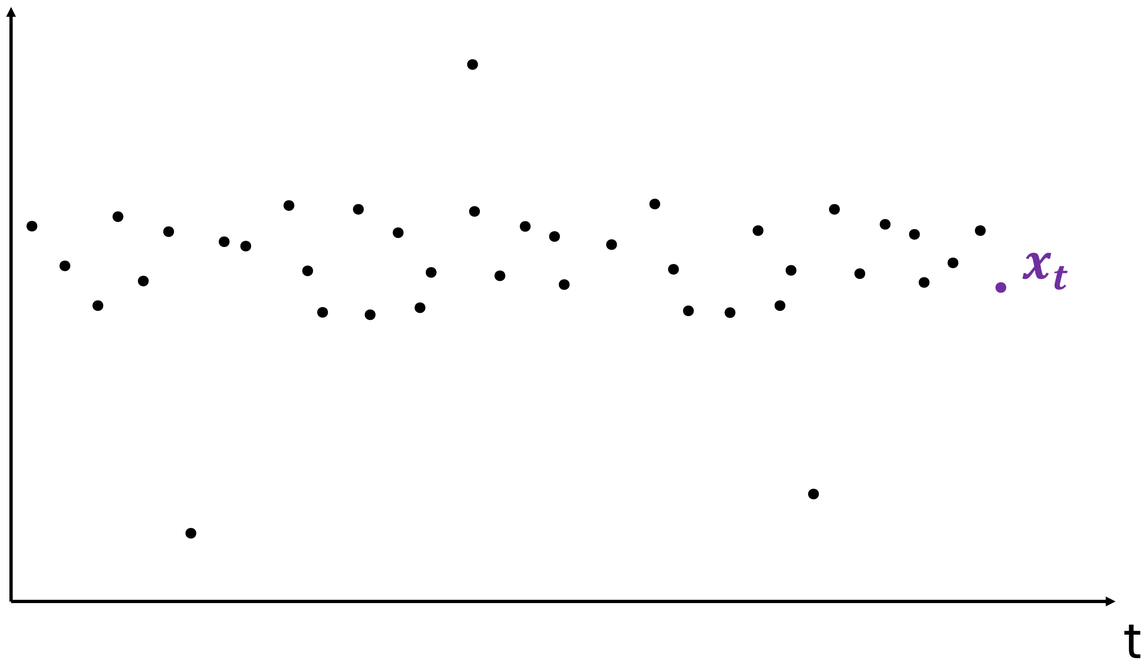
Online Anomaly Detection

Online Anomaly Detection

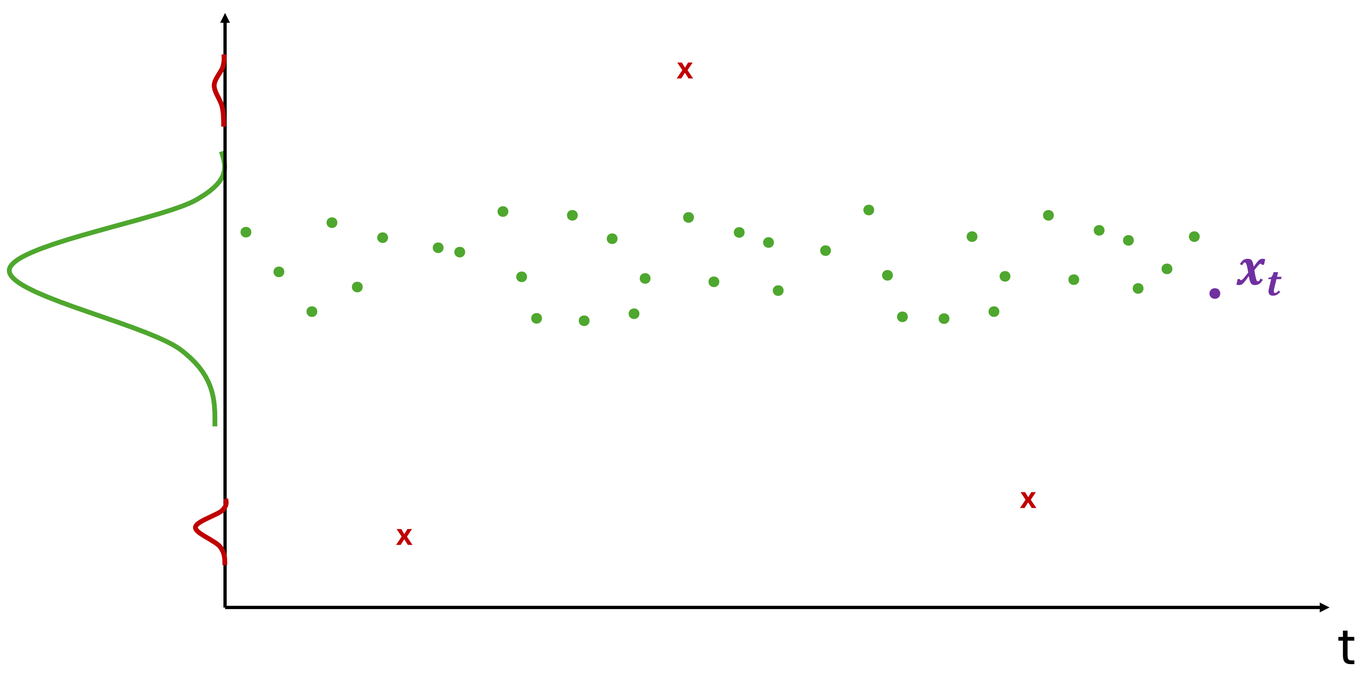
Online Anomaly Detection

Online Anomaly Detection

Given a point x_t

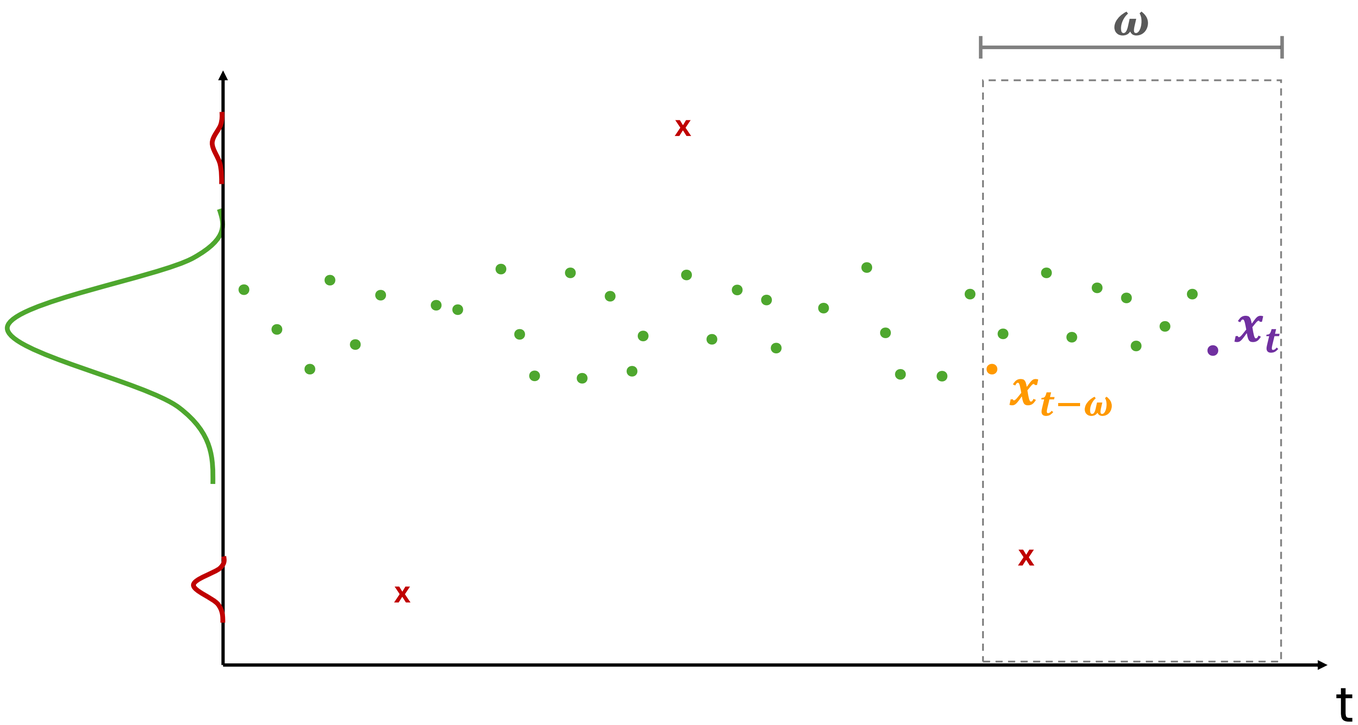


Online Anomaly Detection



Given a point x_t, identify whether it comes from Φ_0 or Φ_1, under the assumption that anomalies are "*few*" and "*different*".

Online Anomaly Detection

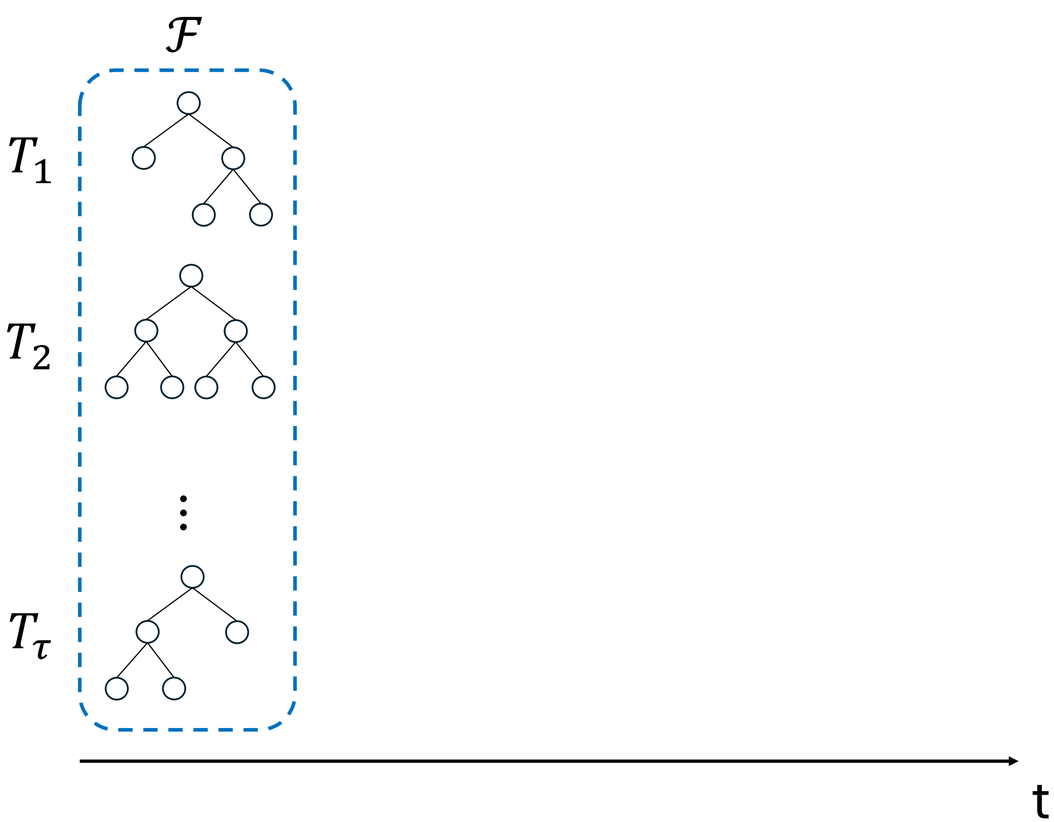


Given a point x_t, identify whether it comes from Φ_0 or Φ_1, under the assumption that anomalies are "*few*" and "*different*".

Requirements:

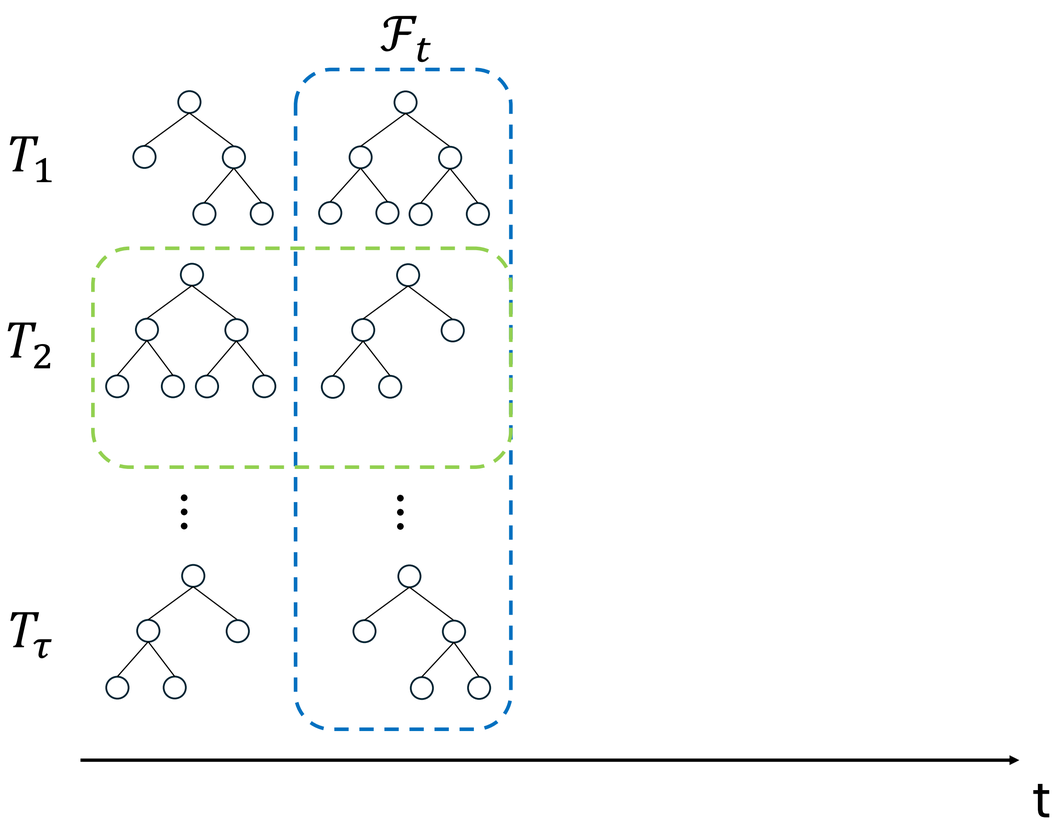
- store in memory only a subset $x_{t-\omega}, \dots, x_t$ at each time instant t.
- classification time of x_t must be as small as possible.

Solution idea



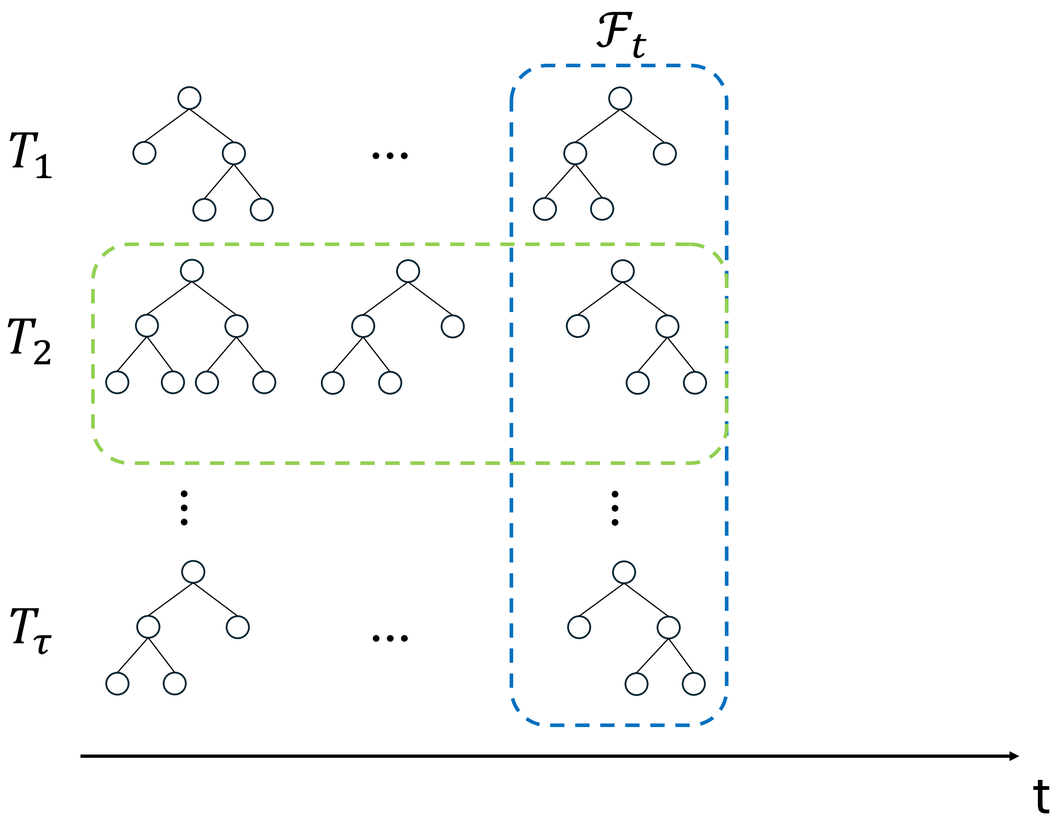
1) Build an ensemble $\mathcal{F}$ of trees

Solution idea



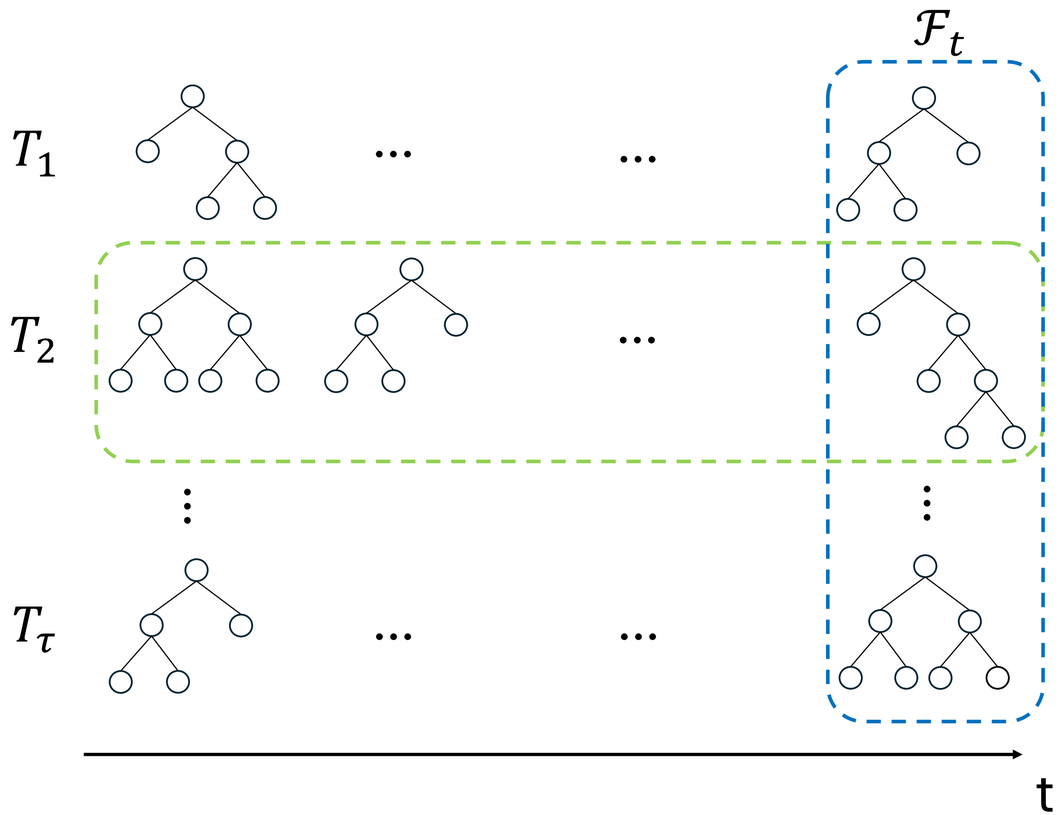
1) Build an ensemble $\mathcal{F}$ of trees that continuously **expand and contract their structure** at each time instant t

Solution idea



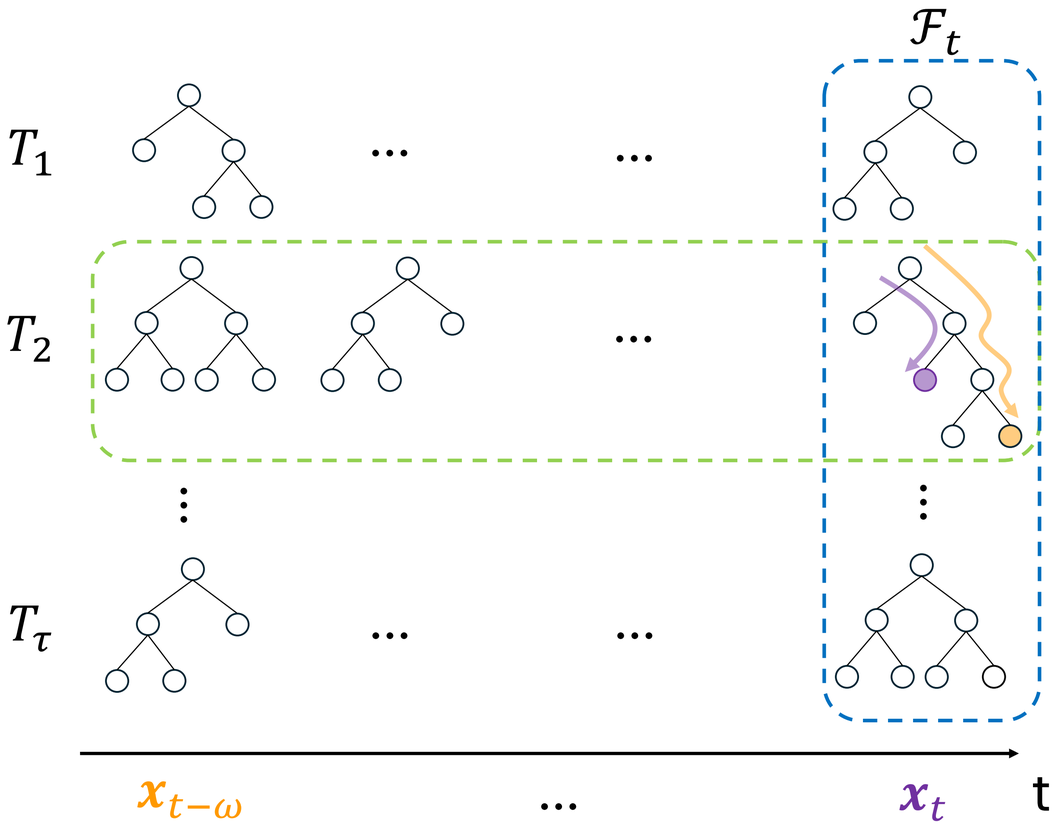
1) Build an ensemble $\mathcal{F}$ of trees that continuously **expand and contract their structure** at each time instant t

Solution idea



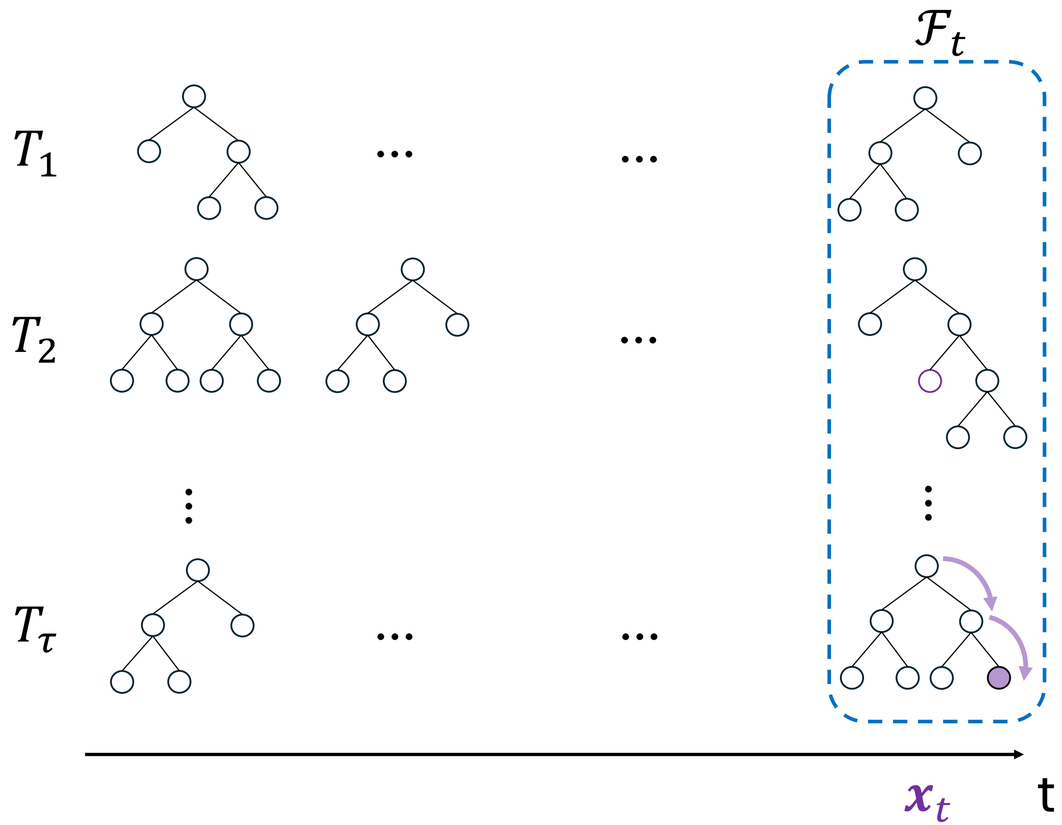
1) Build an ensemble $\mathcal{F}$ of trees that continuously **expand and contract their structure** at each time instant t

Solution idea



1) Build an ensemble $\mathcal{F}$ of trees that continuously **expand and contract their structure** at each time instant t, by learning the new sample $\boldsymbol{x_t}$, and forgetting the old sample $\boldsymbol{x_{t-\omega}}$.

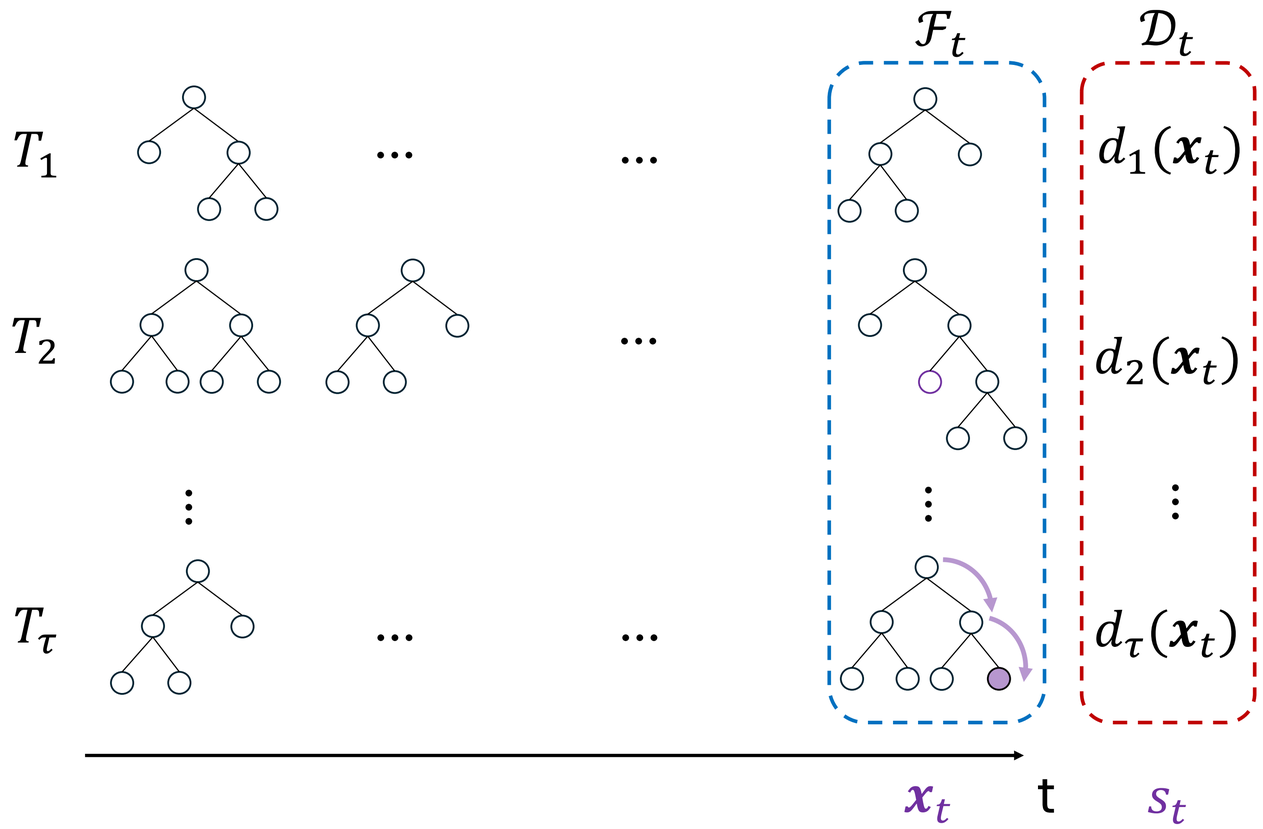
Solution idea 



1) Build an ensemble $\mathcal{F}$ of trees that continuously **expand and contract their structure** at each time instant t, by learning the new sample $\boldsymbol{x}_t$, and forgetting the old sample $\boldsymbol{x}_{t-\omega}$.
2) Compute anomaly score s_t for $\boldsymbol{x}_t$ as a function of the average depth $E(\mathcal{D}_t)$ of all the leaves where $\boldsymbol{x}_t$ falls in each tree:

$$s_t = 2^{-\frac{E(\mathcal{D}_t)}{c(\omega,\eta)}}$$

Solution idea



1) Build an ensemble $\mathcal{F}$ of trees that continuously **expand and contract their structure** at each time instant t, by learning the new sample $\boldsymbol{x}_t$, and forgetting the old sample $\boldsymbol{x}_{t-\omega}$.
2) Compute anomaly score s_t for $\boldsymbol{x}_t$ as a function of the average depth $E(\mathcal{D}_t)$ of all the leaves where $\boldsymbol{x}_t$ falls in each tree:

$$s_t = 2^{-\frac{E(\mathcal{D}_t)}{c(\omega,\eta)}}$$

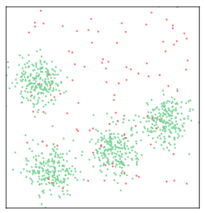
Solution idea

T_1

T_2

T_τ

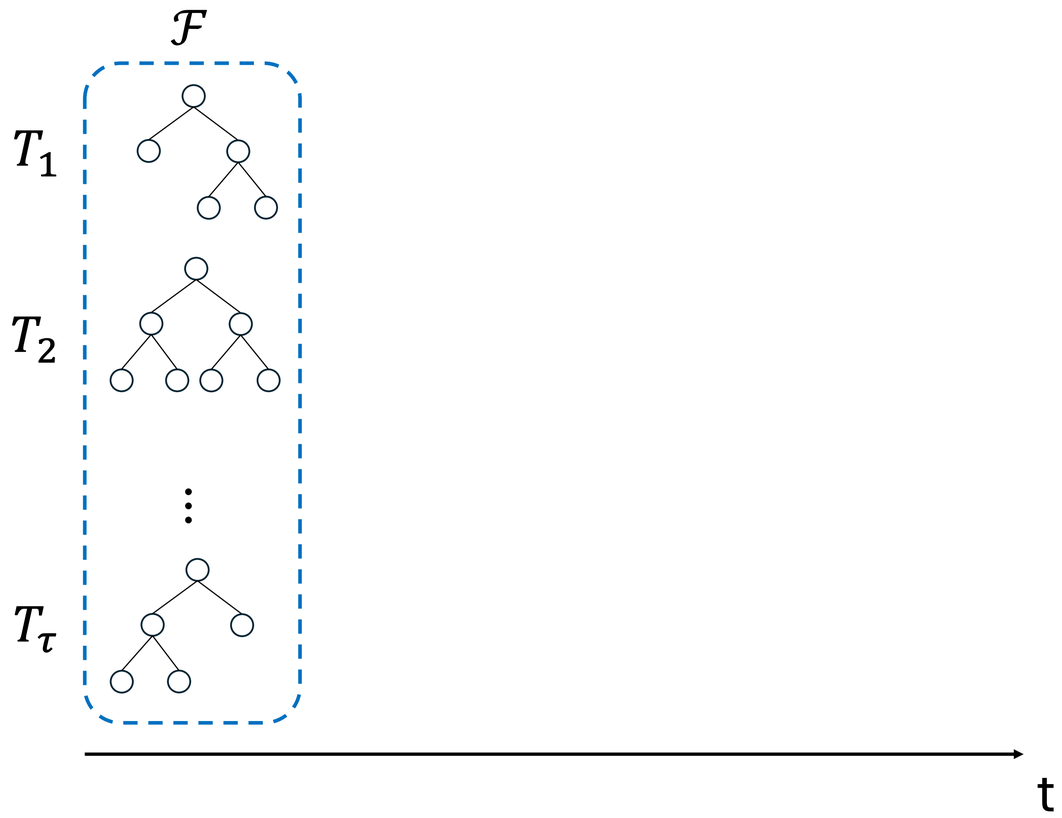
t

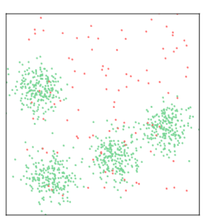


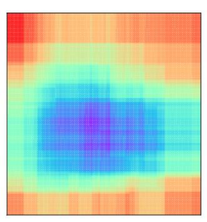
1) Build an ensemble $\mathcal{F}$ of trees that continuously **expand and contract their structure** at each time instant t, by learning the new sample $\boldsymbol{x}_t$, and forgetting the old sample $\boldsymbol{x}_{t-\omega}$.
2) Compute anomaly score s_t for $\boldsymbol{x}_t$ as a function of the average depth $E(\mathcal{D}_t)$ of all the leaves where $\boldsymbol{x}_t$ falls in each tree:

$$s_t = 2^{-\frac{E(\mathcal{D}_t)}{c(\omega,\eta)}}$$

Solution idea 



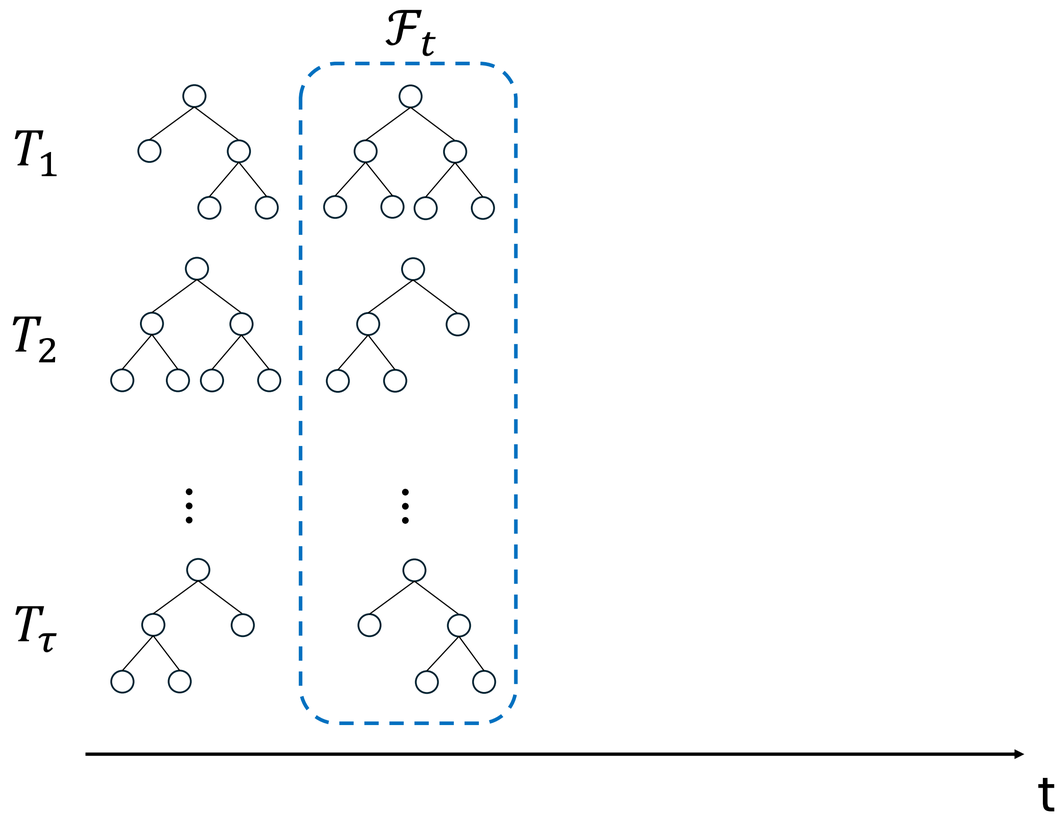


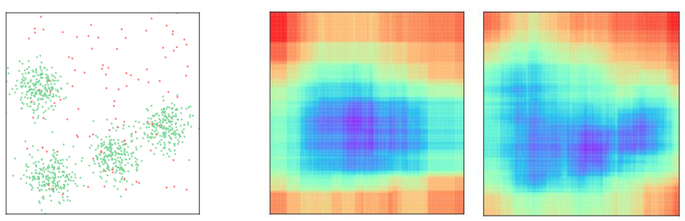


1) Build an ensemble $\mathcal{F}$ of trees that continuously **expand and contract their structure** at each time instant t, by learning the new sample $\boldsymbol{x}_t$, and forgetting the old sample $\boldsymbol{x}_{t-\omega}$.
2) Compute anomaly score s_t for $\boldsymbol{x}_t$ as a function of the average depth $E(\mathcal{D}_t)$ of all the leaves where $\boldsymbol{x}_t$ falls in each tree:

$$s_t = 2^{-\frac{E(\mathcal{D}_t)}{c(\omega,\eta)}}$$

Solution idea

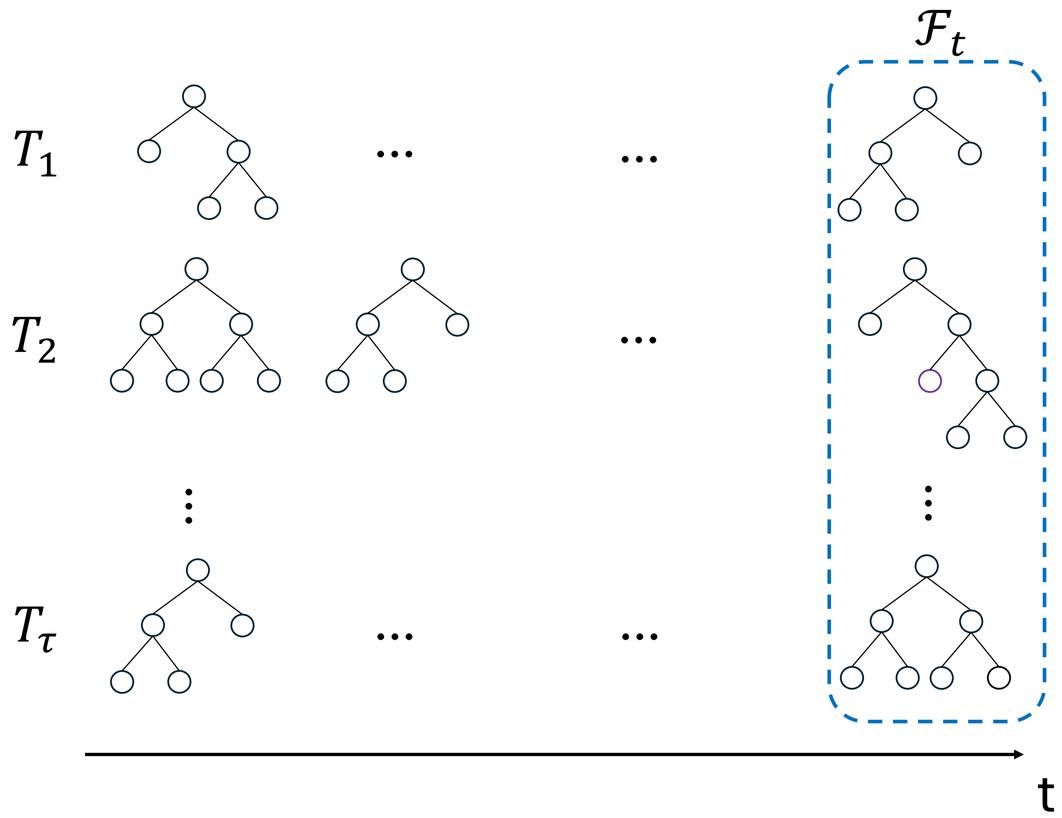


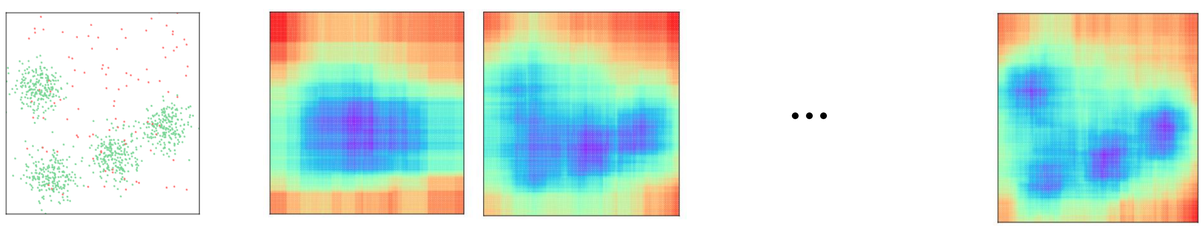


1) Build an ensemble $\mathcal{F}$ of trees that continuously **expand and contract their structure** at each time instant t, by learning the new sample $\boldsymbol{x}_t$, and forgetting the old sample $\boldsymbol{x}_{t-\omega}$.
2) Compute anomaly score s_t for $\boldsymbol{x}_t$ as a function of the average depth $E(\mathcal{D}_t)$ of all the leaves where $\boldsymbol{x}_t$ falls in each tree:

$$s_t = 2^{-\frac{E(\mathcal{D}_t)}{c(\omega,\eta)}}$$

Solution idea

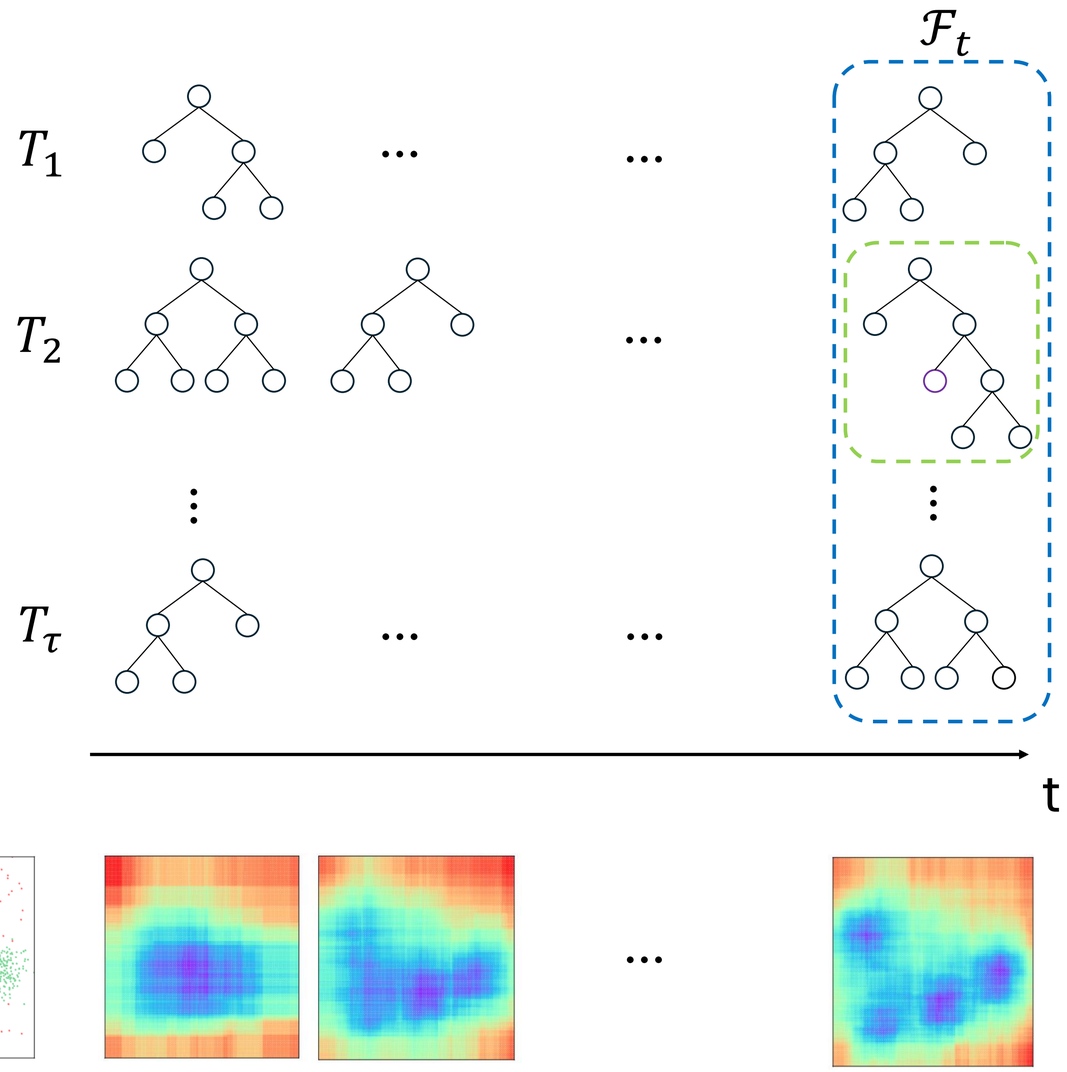




1) Build an ensemble $\mathcal{F}$ of trees that continuously **expand and contract their structure** at each time instant t, by learning the new sample $\boldsymbol{x}_t$, and forgetting the old sample $\boldsymbol{x}_{t-\omega}$.
2) Compute anomaly score s_t for $\boldsymbol{x}_t$ as a function of the average depth $E(\mathcal{D}_t)$ of all the leaves where $\boldsymbol{x}_t$ falls in each tree:

$$s_t = 2^{-\frac{E(\mathcal{D}_t)}{c(\omega,\eta)}}$$

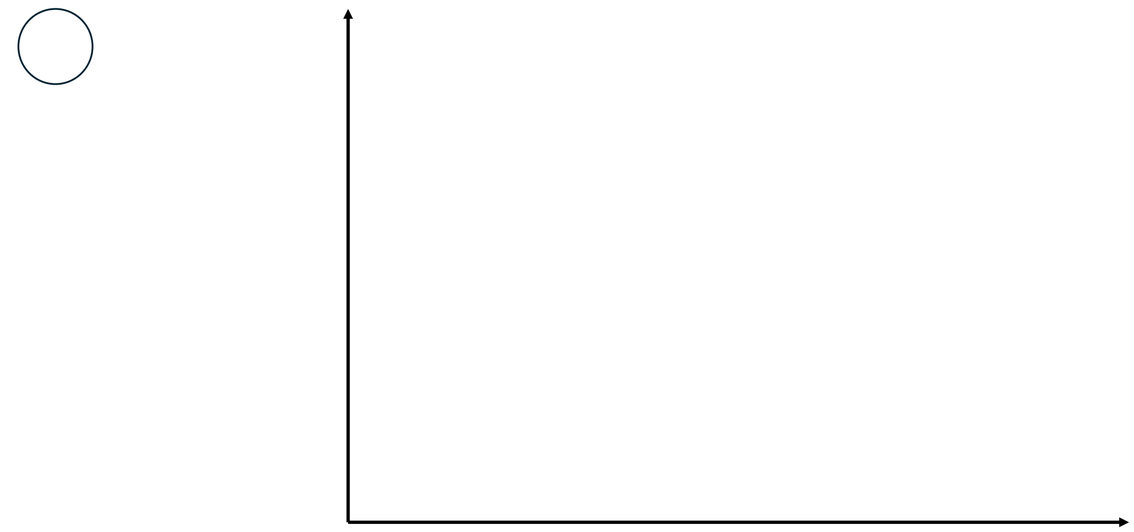
Solution idea



1) Build an ensemble $\mathcal{F}$ of trees that continuously **expand and contract their structure** at each time instant t, by learning the new sample $\boldsymbol{x}_t$, and forgetting the old sample $\boldsymbol{x}_{t-\omega}$.
2) Compute anomaly score s_t for $\boldsymbol{x}_t$ as a function of the average depth $E(\mathcal{D}_t)$ of all the leaves where $\boldsymbol{x}_t$ falls in each tree:

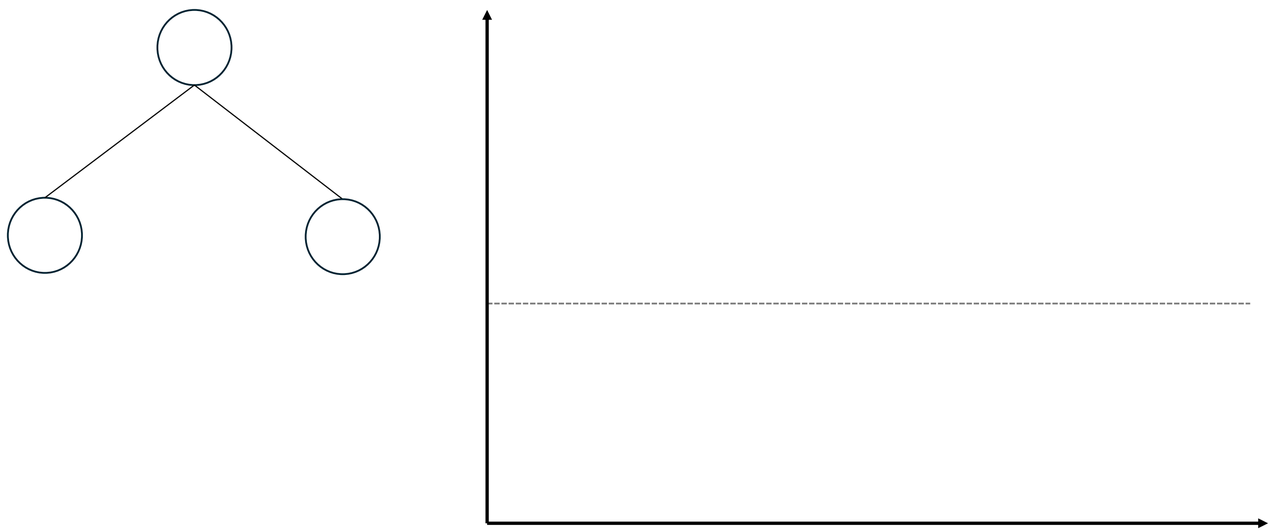
$$s_t = 2^{-\frac{E(\mathcal{D}_t)}{c(\omega,\eta)}}$$

Online Isolation Tree



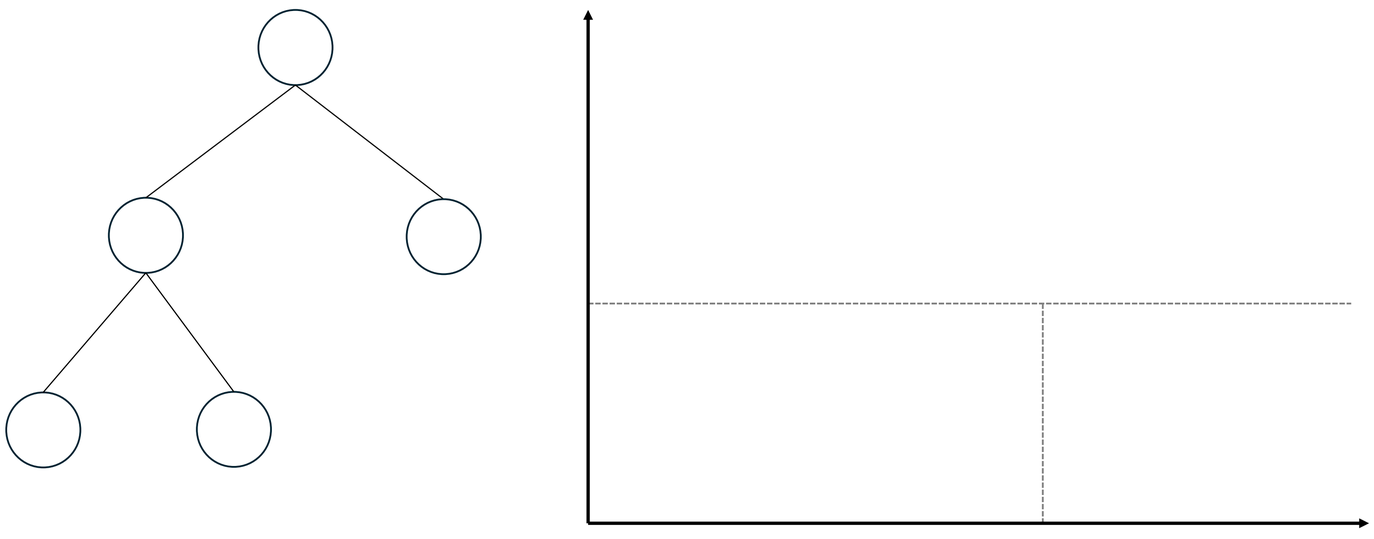
Online Isolation Tree is a ***d*-dimensional histogram** built by recursively splitting the input space $\mathbb{R}^d$.

Online Isolation Tree



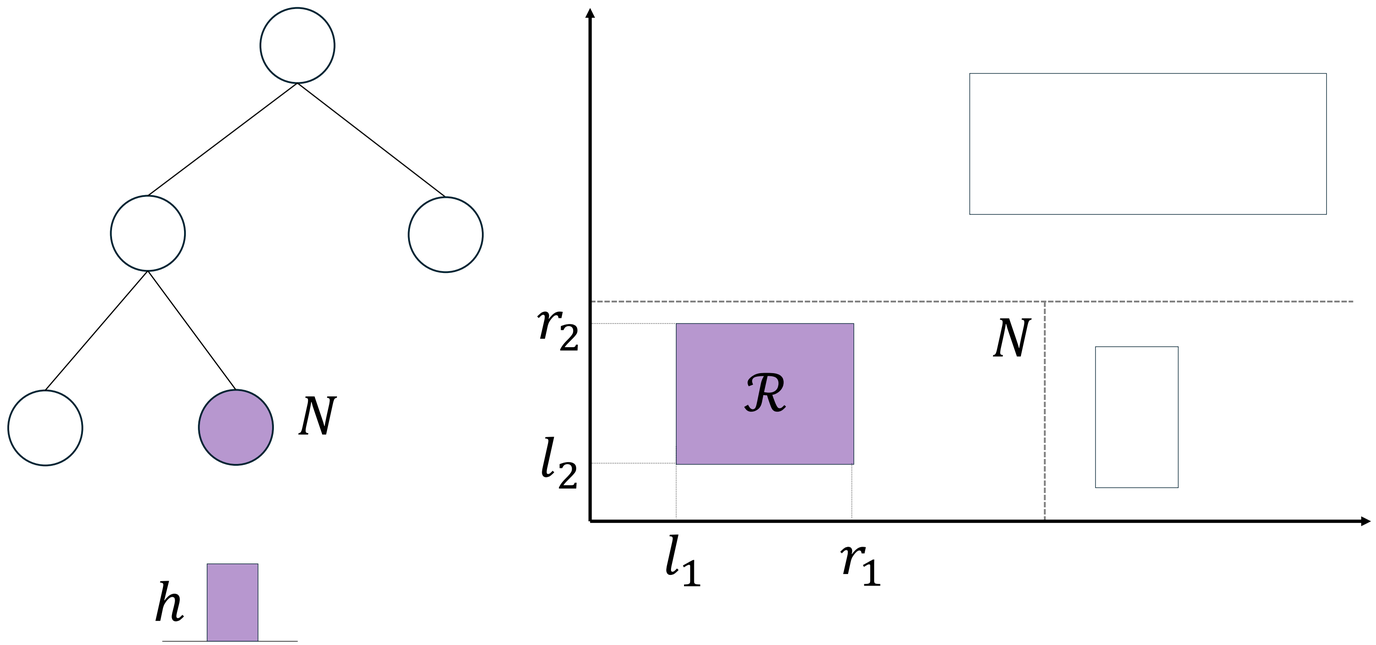
Online Isolation Tree is a ***d*-dimensional histogram** built by recursively splitting the input space $\mathbb{R}^d$.

Online Isolation Tree



Online Isolation Tree is a ***d*-dimensional histogram** built by recursively splitting the input space $\mathbb{R}^d$.

Online Isolation Tree

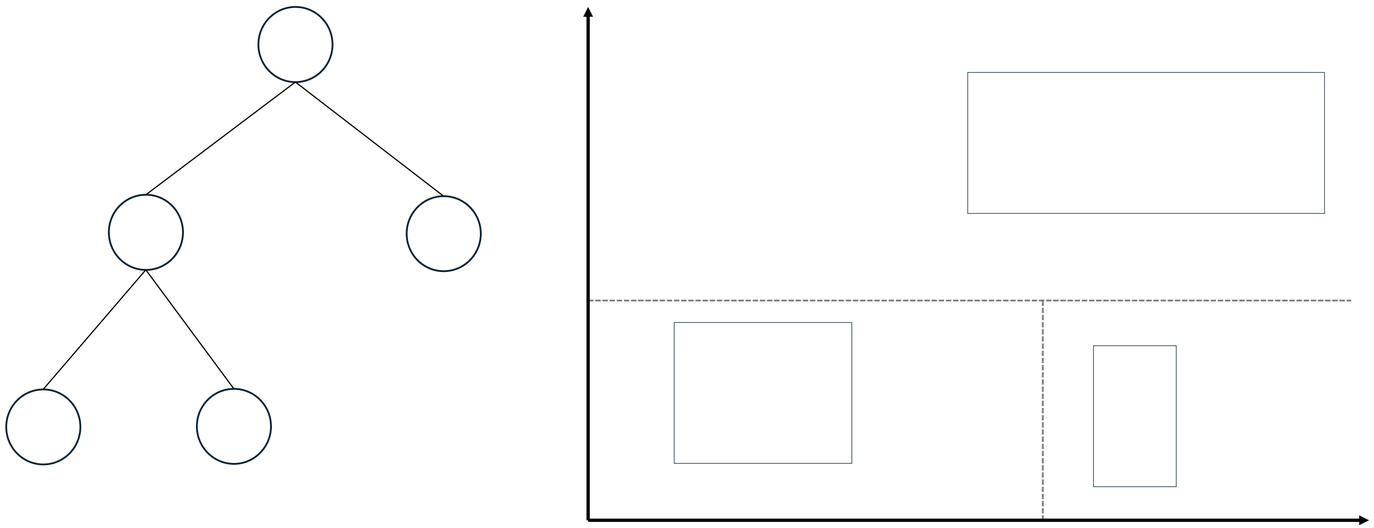


Online Isolation Tree is a ***d*-dimensional histogram** built by recursively splitting the input space $\mathbb{R}^d$.

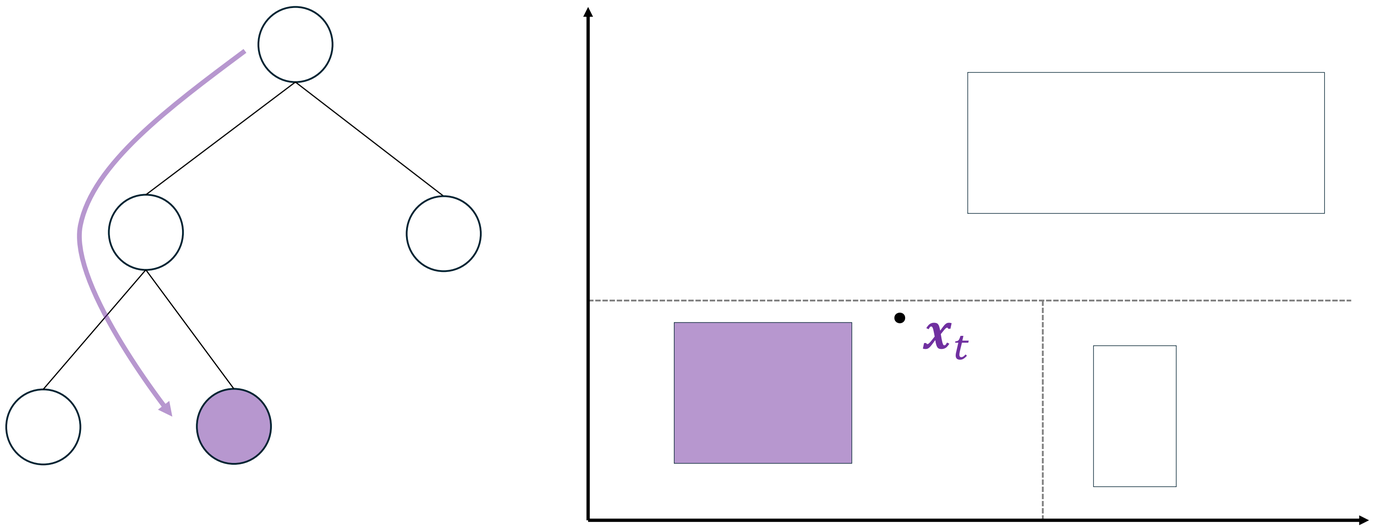
Each node N is characterized as $N = (h, \mathcal{R})$, where:

- h is the number of points that crossed it, that is the bin height,
- $\mathcal{R} = \times_{i=1}^{d} [l_i, r_i]$ is the minimal bounding box that encloses them, that is the support of the bin.

Learning procedure

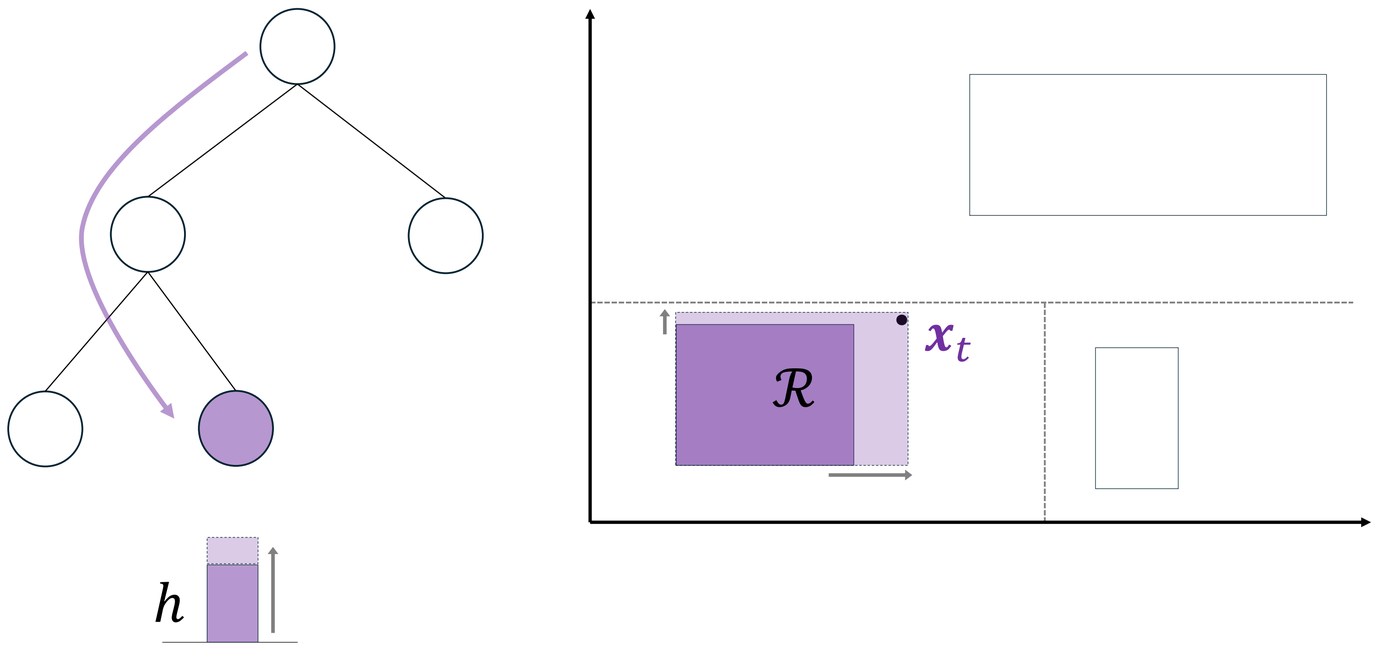


Learning procedure



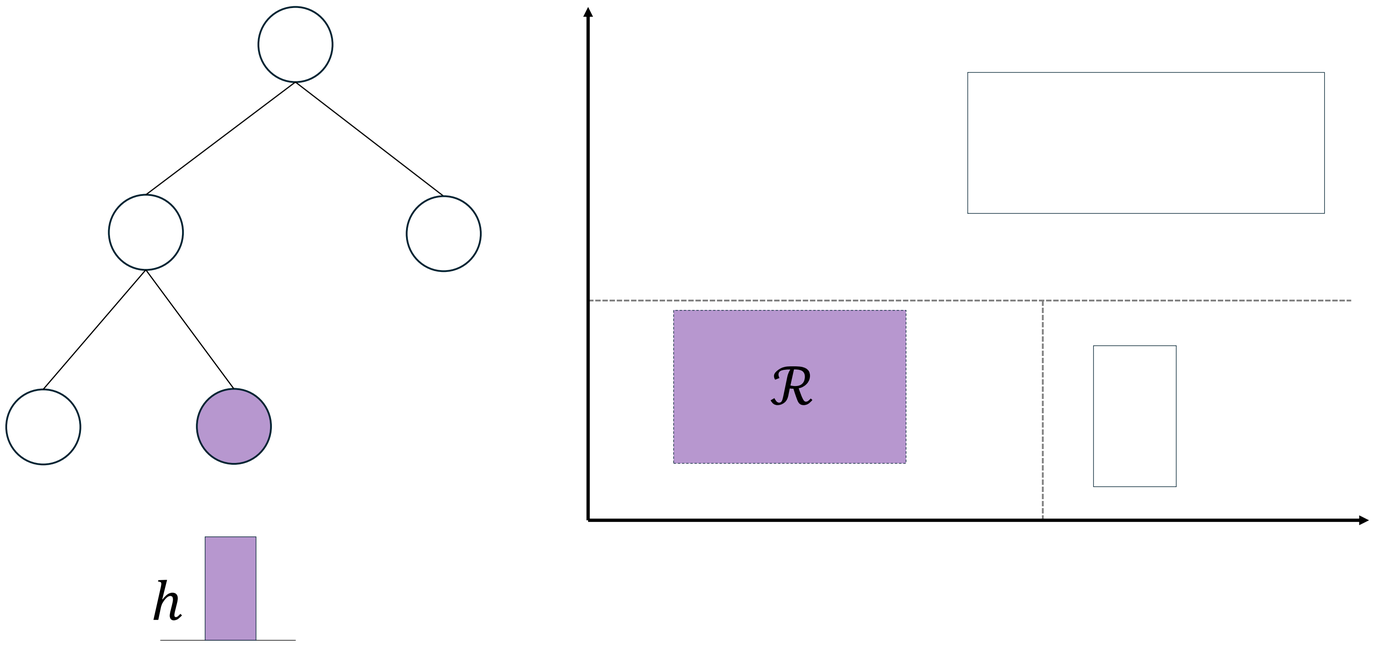
At each new point $\boldsymbol{x}_t$

Learning procedure



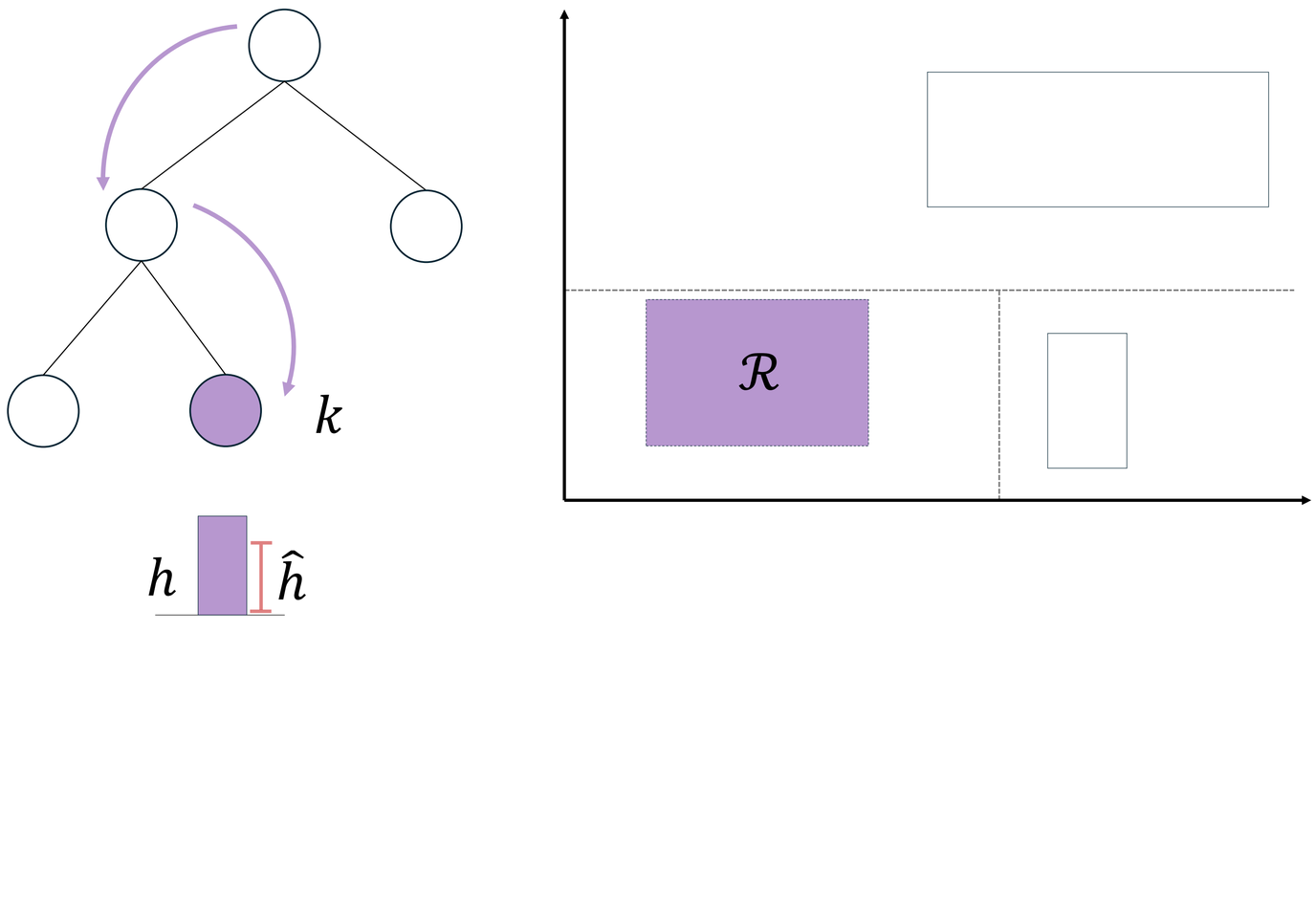
At each new point x_t, the heights h and supports $\mathcal{R}$ are updated.

Learning procedure



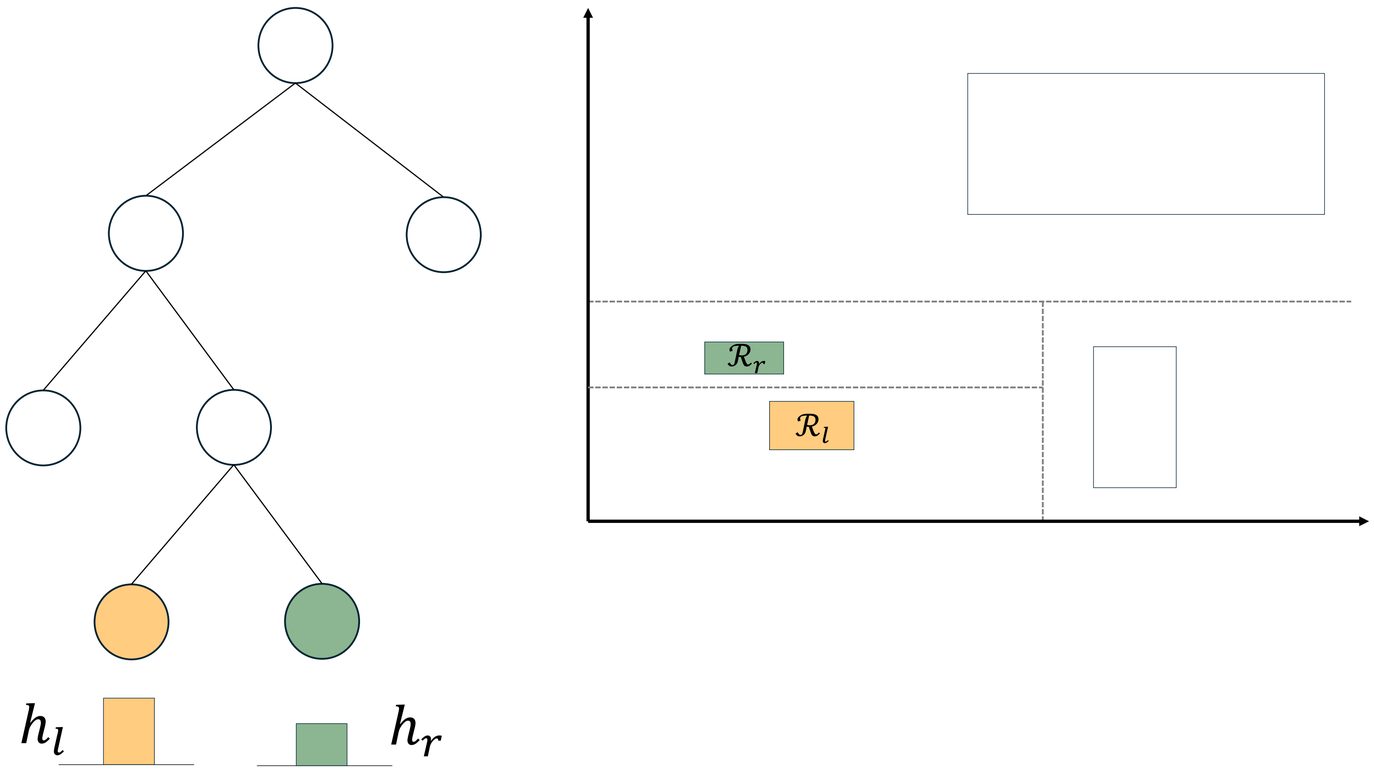
At each new point x_t, the heights h and supports $\mathcal{R}$ are updated.

Learning procedure



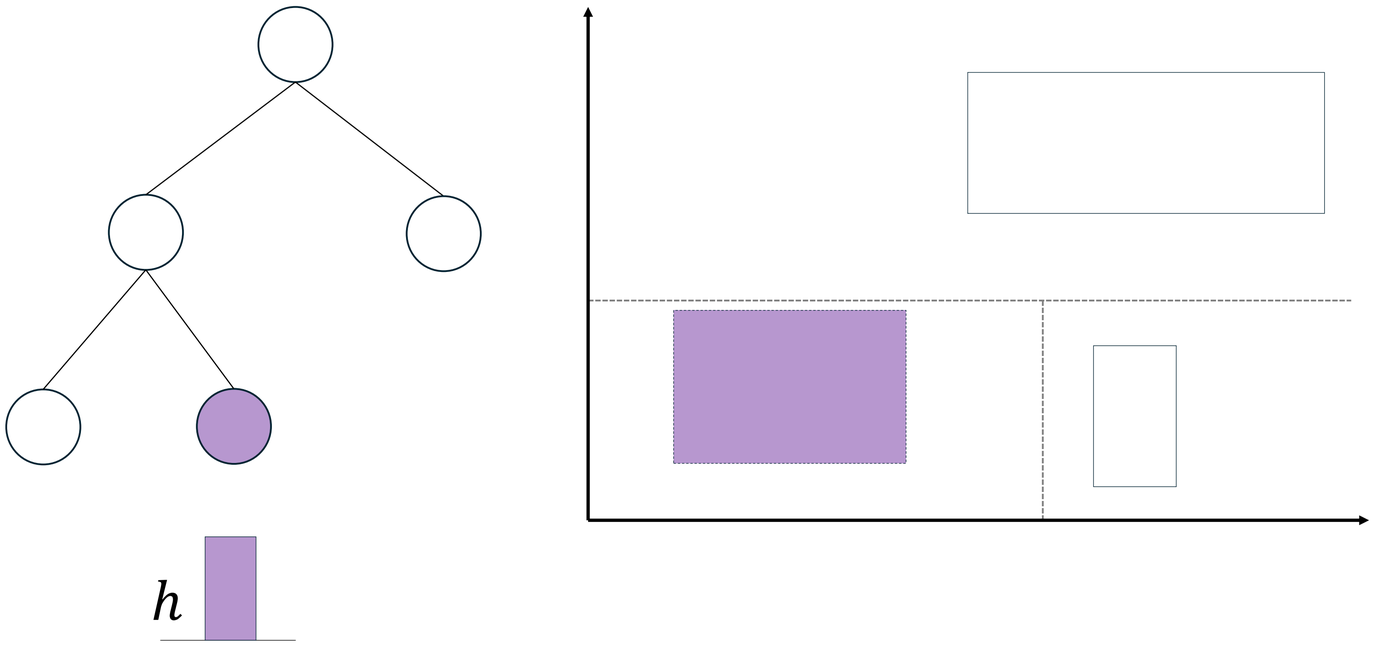
At each new point $\boldsymbol{x}_t$, the heights h and supports $\mathcal{R}$ are updated.
If the leaf height h reaches the maximum value $\hat{h} = \eta 2^k$

Learning procedure

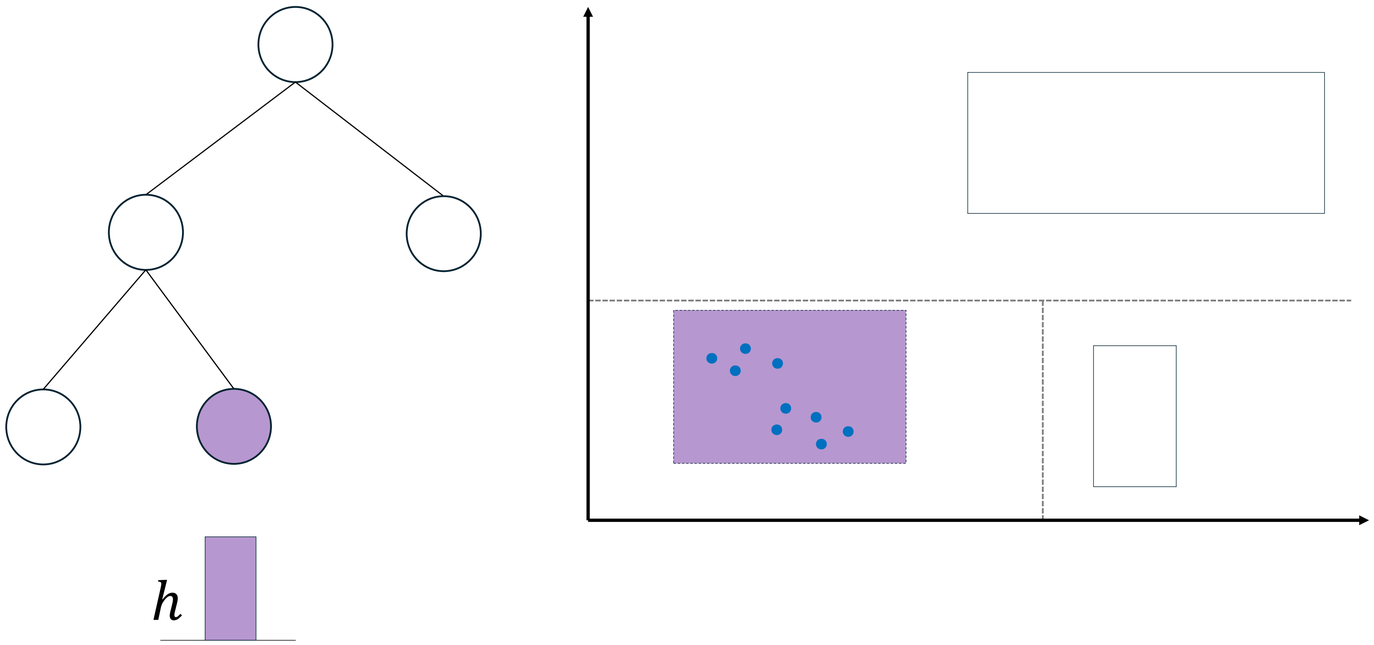


At each new point $\boldsymbol{x}_t$, the heights h and supports $\mathcal{R}$ are updated.
If the leaf height h reaches the maximum value $\hat{h} = \eta 2^k$, we split the corresponding bin, leading to a **tree expansion**.

Learning procedure

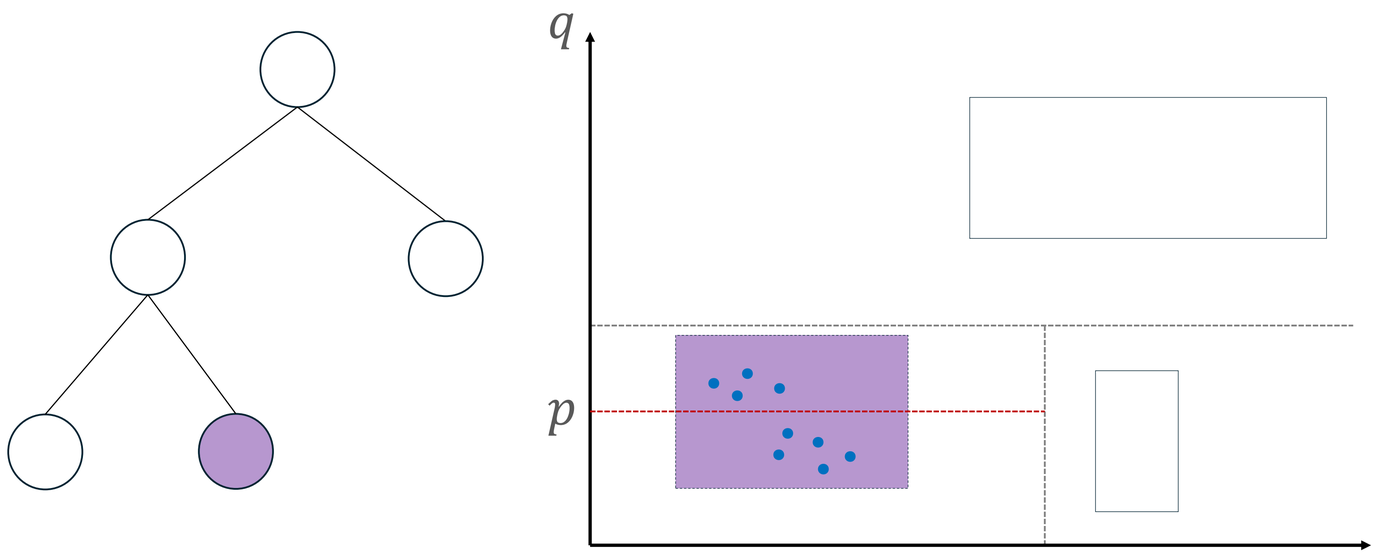


Learning procedure



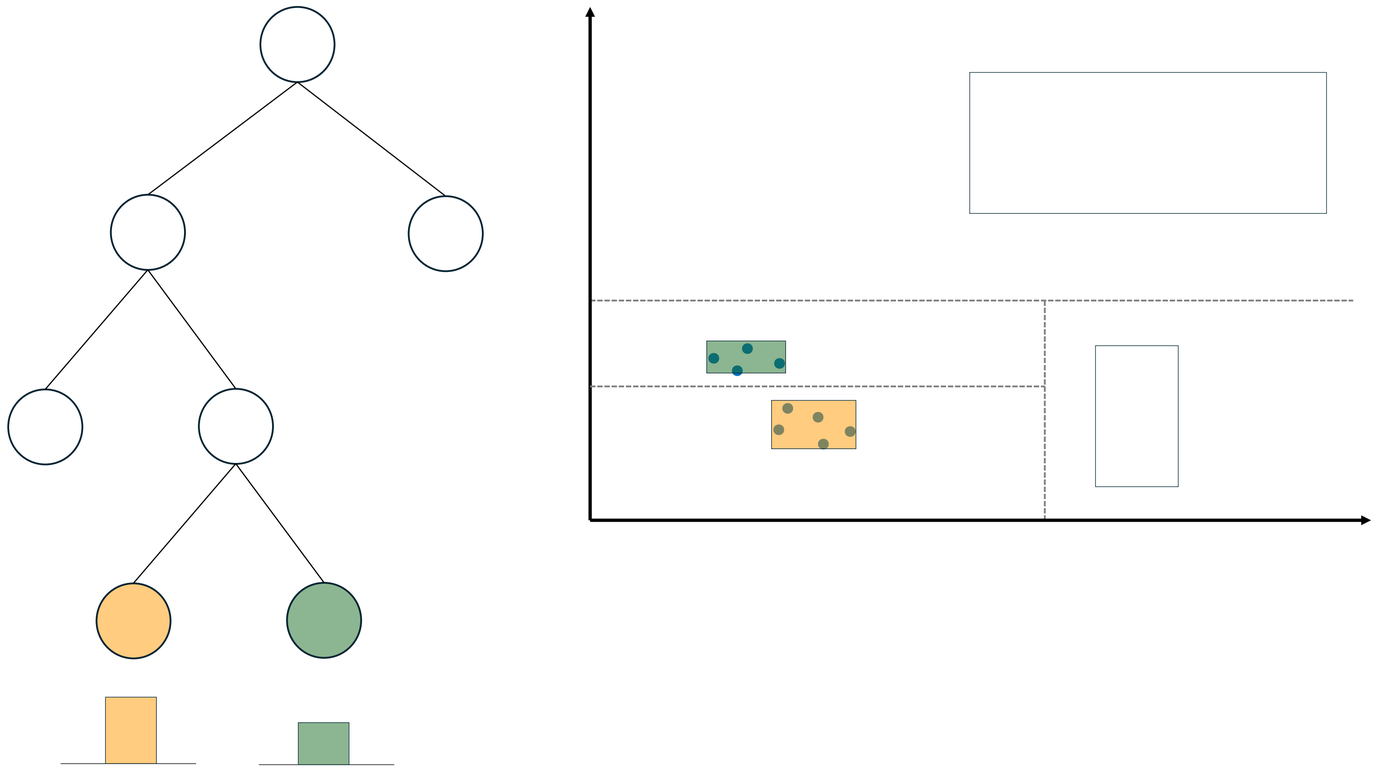
1. Randomly sample h points $\boldsymbol{x} \sim \mathcal{U}_{\mathcal{R}}$,

Learning procedure



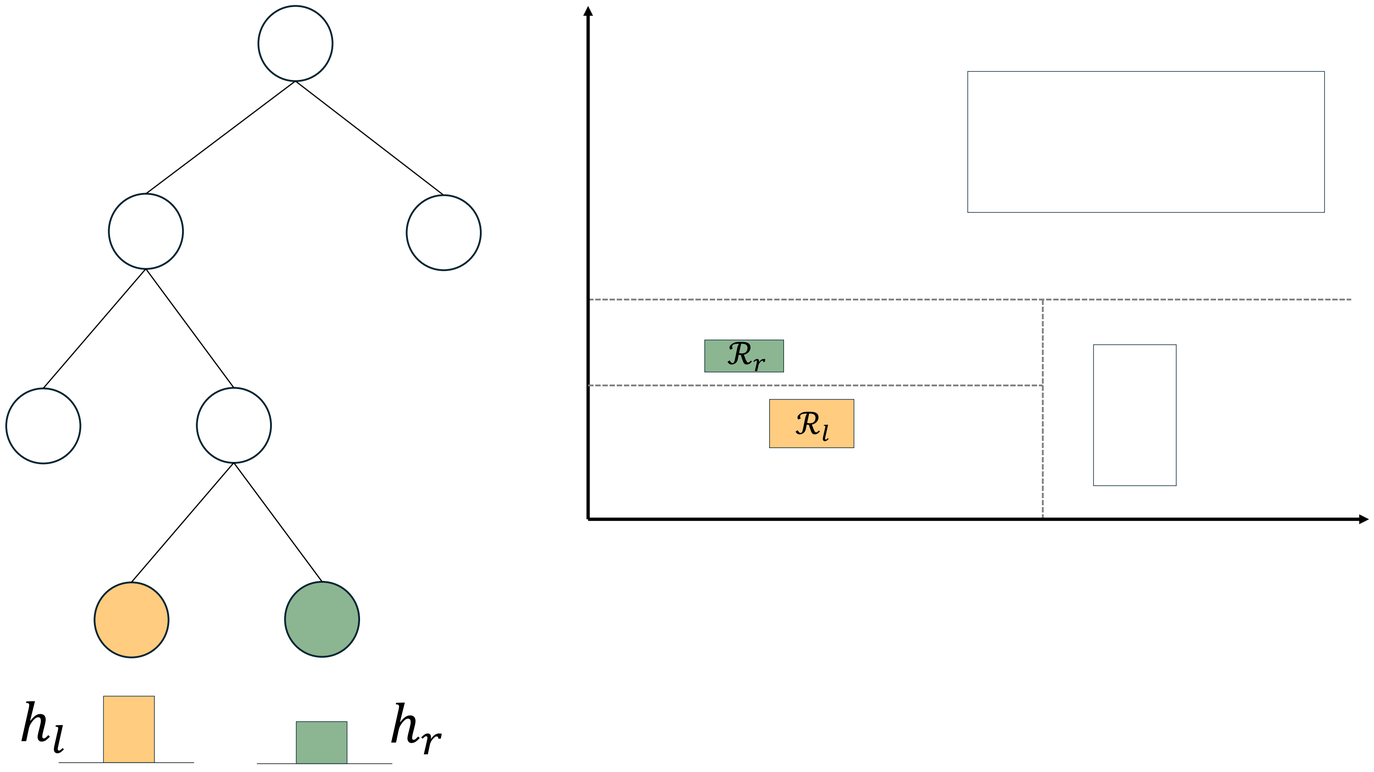
1. Randomly sample h points $\boldsymbol{x} \sim \mathcal{U}_{\mathcal{R}}$,
2. Randomly sample a dimension $q \sim \mathcal{U}_{\{1,\dots,d\}}$ and split value $p \sim \mathcal{U}_{[l_q, r_q]}$,

Learning procedure



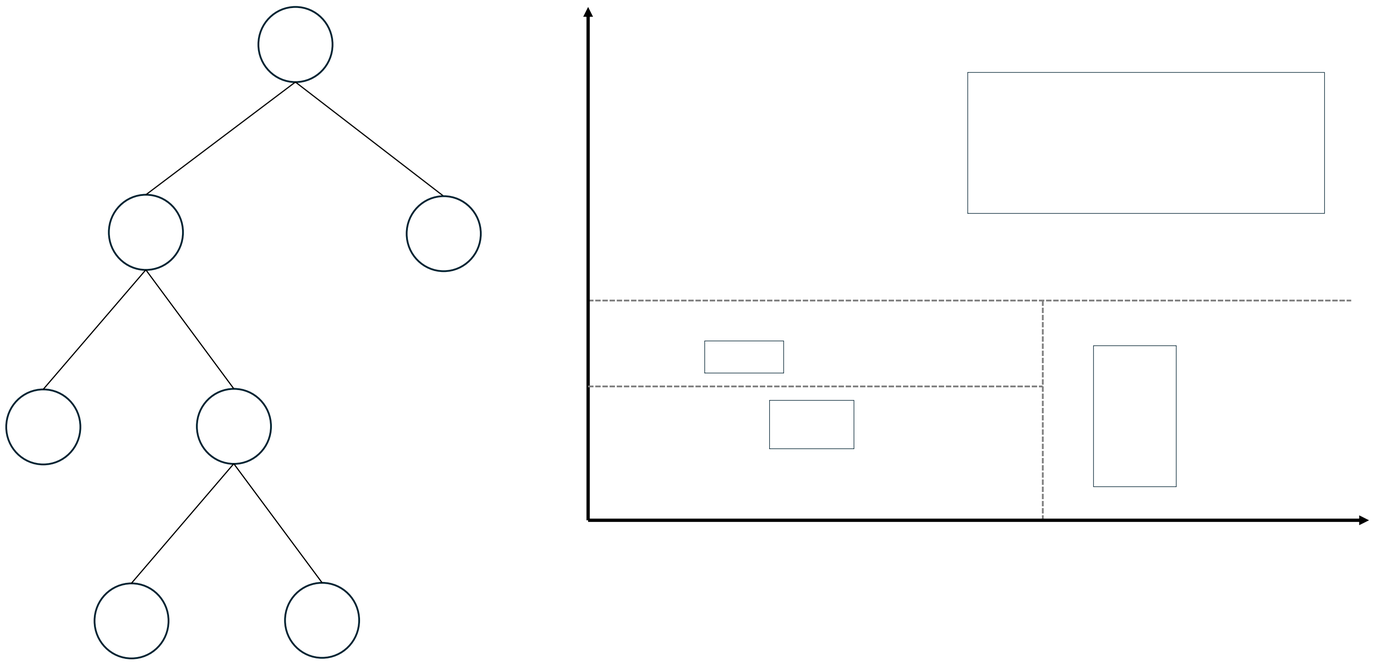
1. Randomly sample h points $\boldsymbol{x} \sim \mathcal{U}_{\mathcal{R}}$,
2. Randomly sample a dimension $q \sim \mathcal{U}_{\{1,\dots,d\}}$ and split value $p \sim \mathcal{U}_{[l_q, r_q]}$,
3. Initialize heights h_l, h_r and supports $\mathcal{R}_l$, $\mathcal{R}_r$ of left and right child nodes N_l and N_r.

Learning procedure

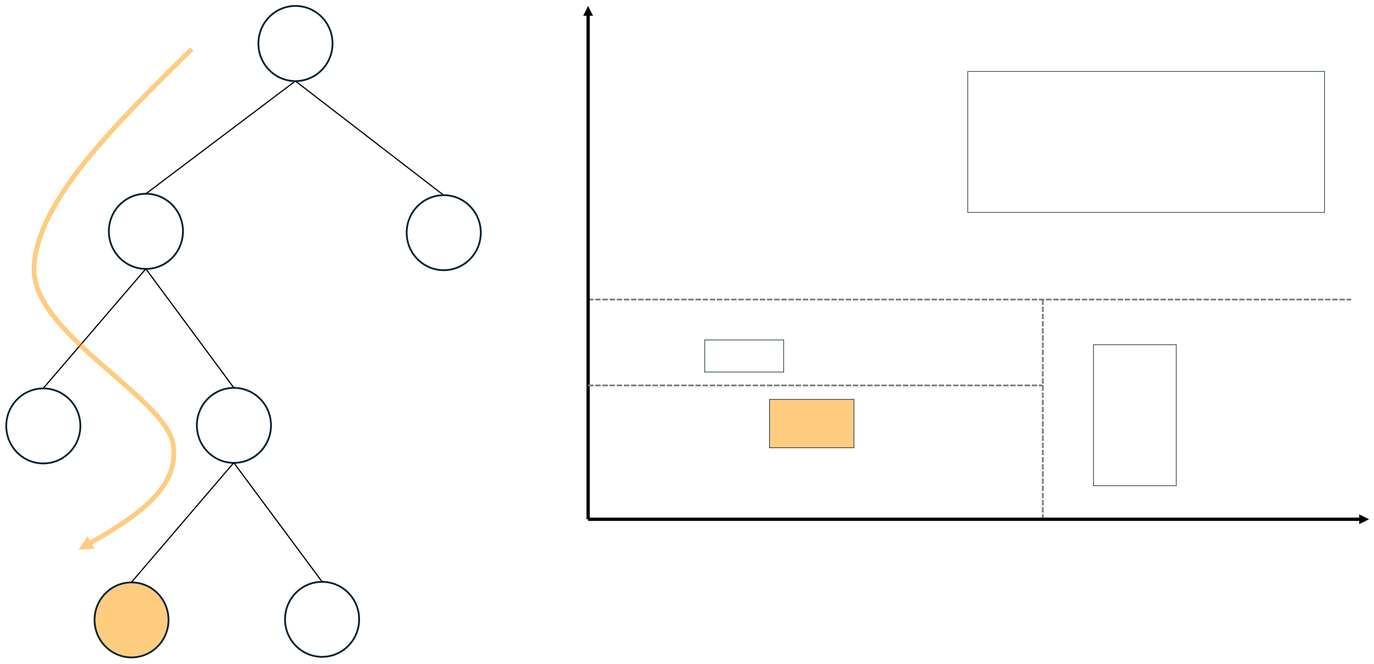


1. Randomly sample h points $\boldsymbol{x} \sim \mathcal{U}_{\mathcal{R}}$,
2. Randomly sample a dimension $q \sim \mathcal{U}_{\{1,\dots,d\}}$ and split value $p \sim \mathcal{U}_{[l_q, r_q]}$,
3. Initialize heights h_l, h_r and supports $\mathcal{R}_l$, $\mathcal{R}_r$ of left and right child nodes N_l and N_r.

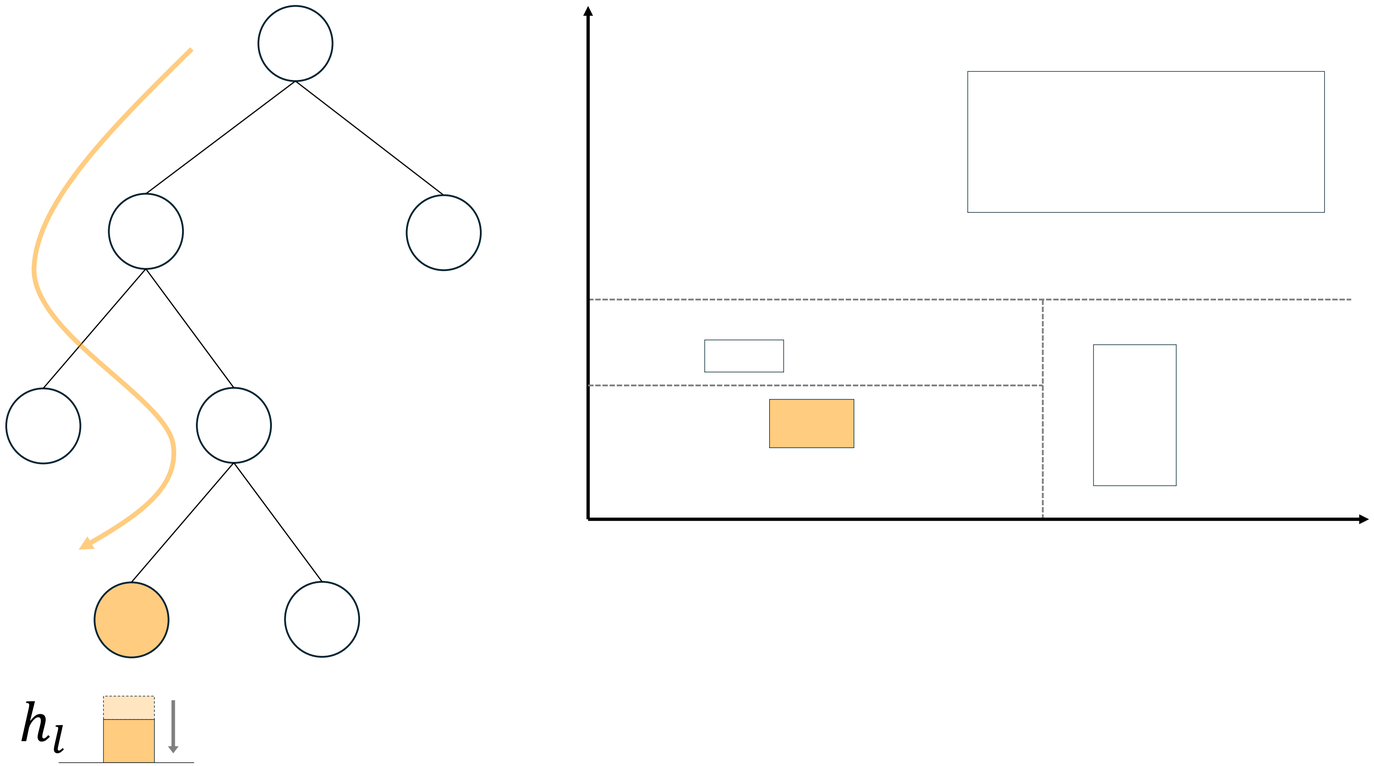
Forgetting procedure



Forgetting procedure

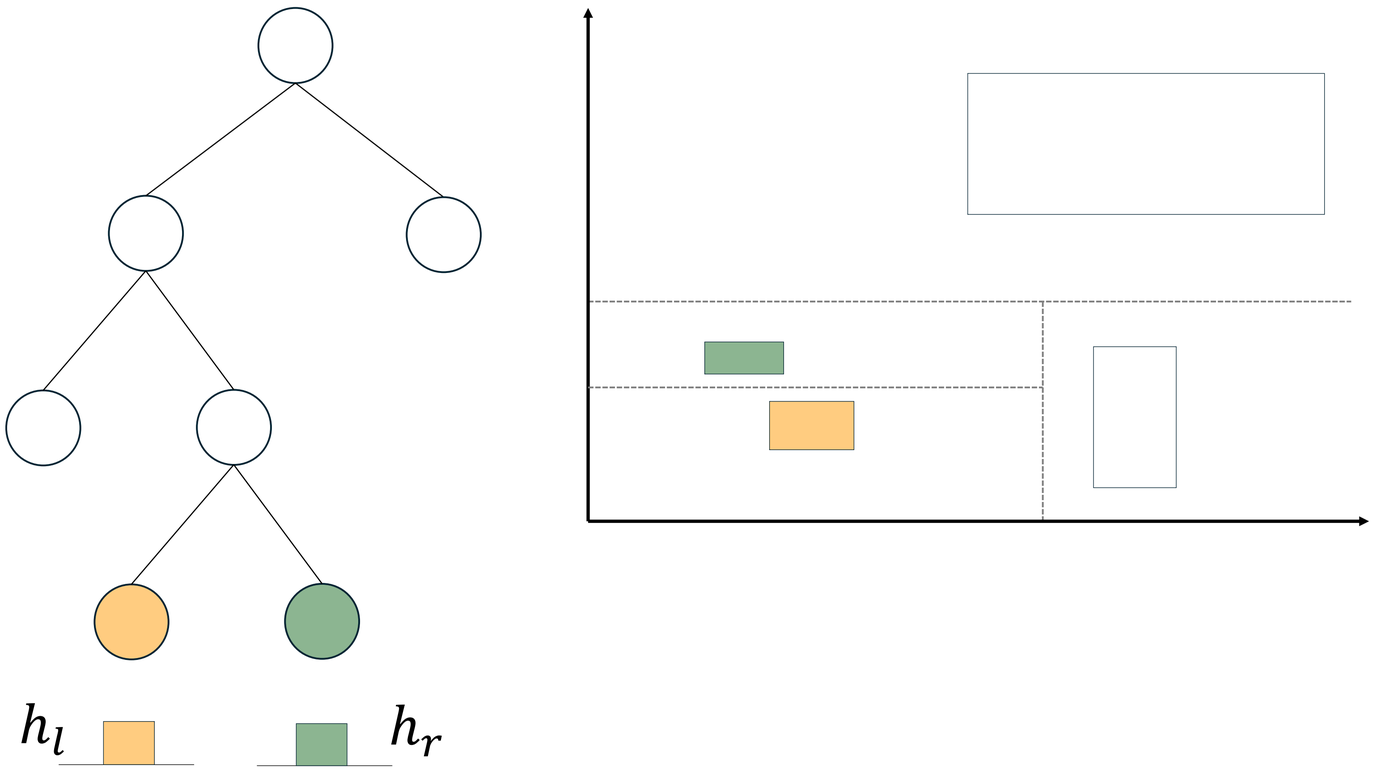


Forgetting procedure



Bin heights h are decreased along the path of $\boldsymbol{x}_{t-\omega}$.

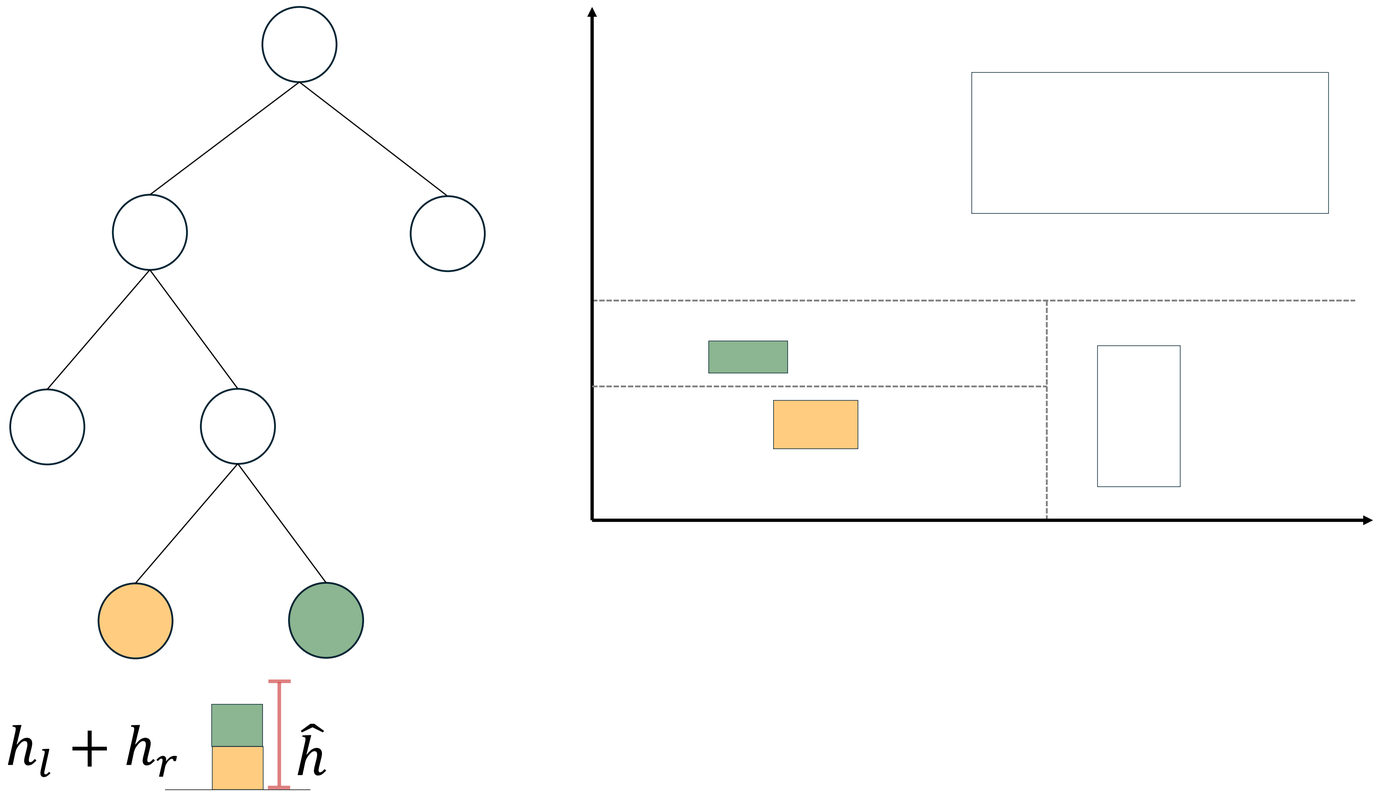
Forgetting procedure



Bin heights h are decreased along the path of $\boldsymbol{x}_{t-\omega}$.

If the sum of two child nodes heights $h_l + h_r$ drops below the value $\hat{h}$, we merge the corresponding bins, leading to a **tree contraction**.

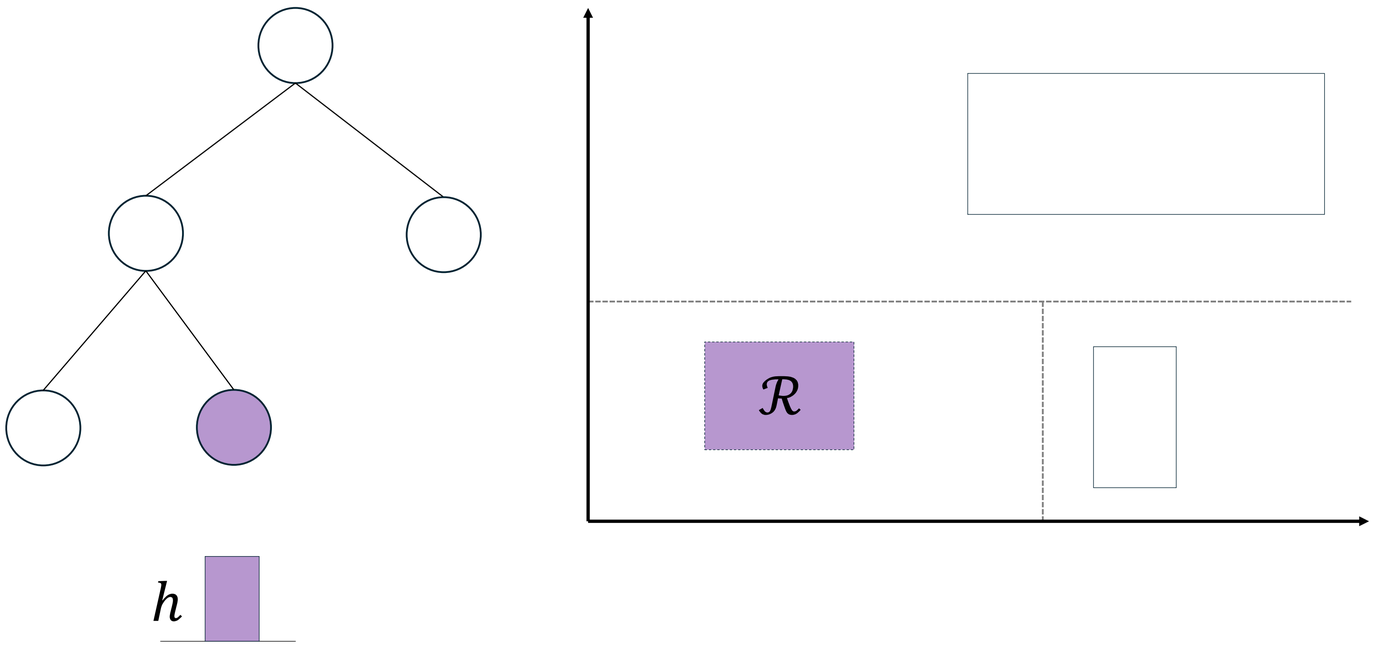
Forgetting procedure



Bin heights h are decreased along the path of $\boldsymbol{x}_{t-\omega}$.

If the sum of two child nodes heights $h_l + h_r$ drops below the value $\hat{h}$, we merge the corresponding bins, leading to a **tree contraction**.

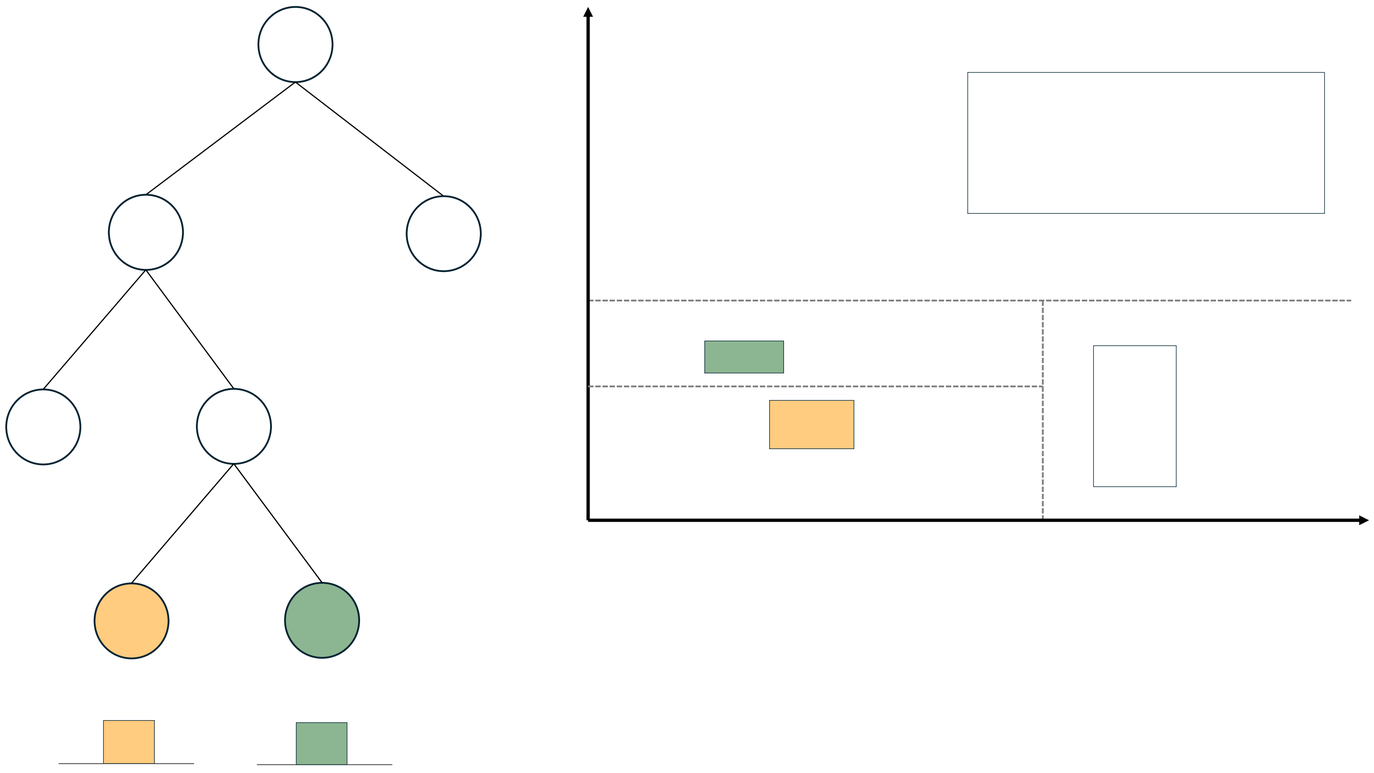
Forgetting procedure



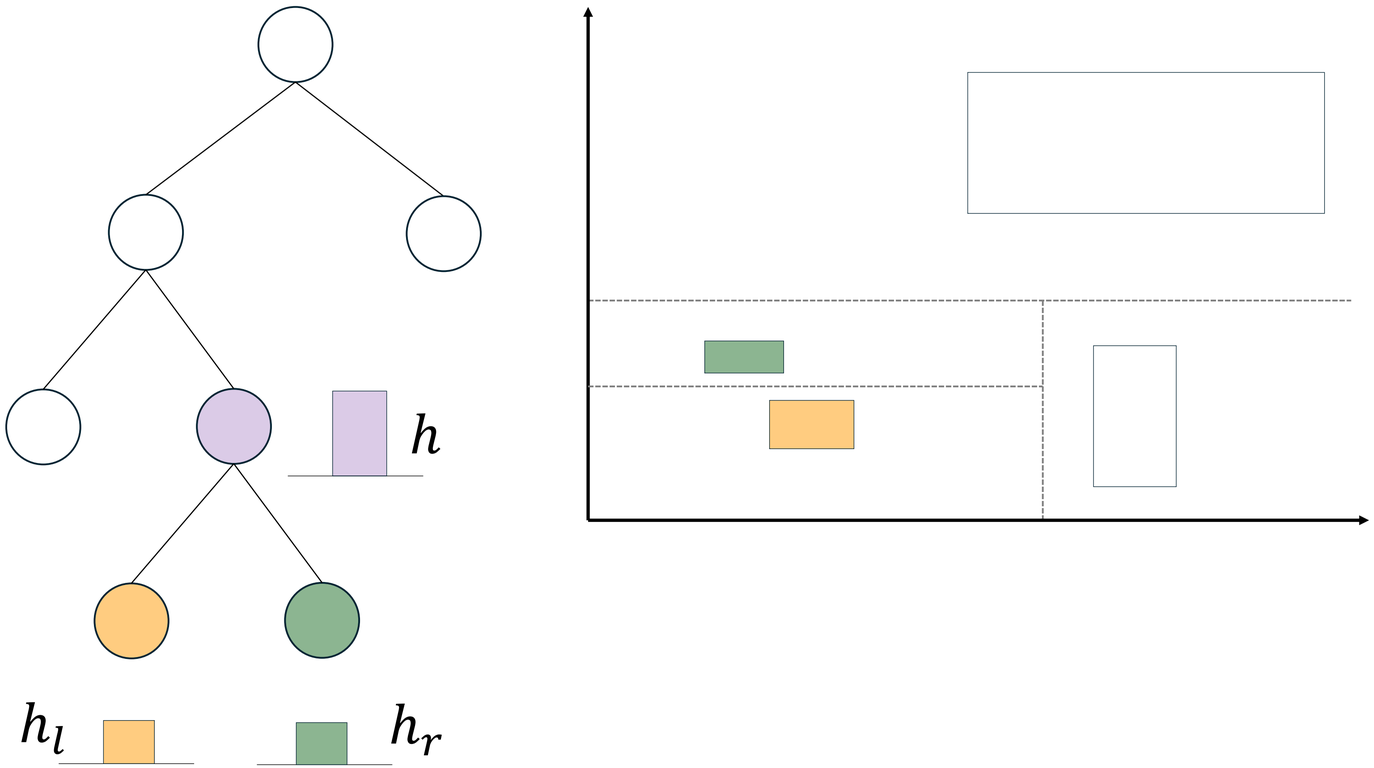
Bin heights h are decreased along the path of $\boldsymbol{x}_{t-\omega}$.

If the sum of two child nodes heights $h_l + h_r$ drops below the value $\hat{h}$, we merge the corresponding bins, leading to a **tree contraction**.

Forgetting procedure

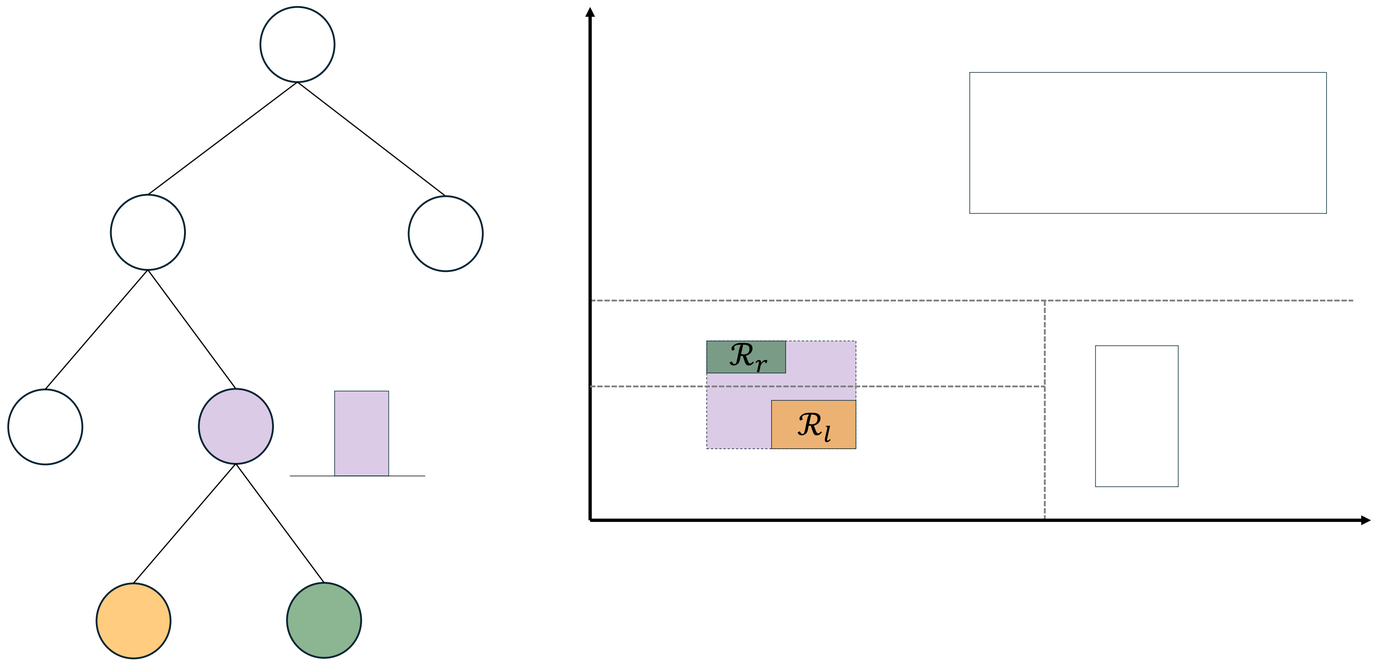


Forgetting procedure



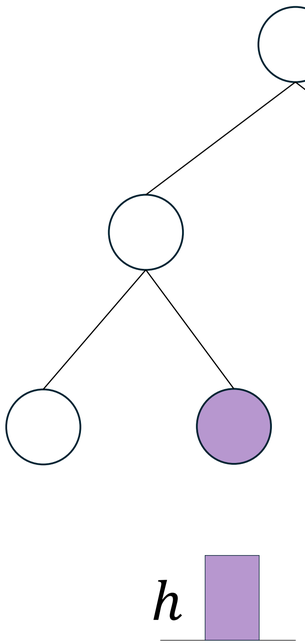
1. Set parent height h as the sum of h_l and h_r,

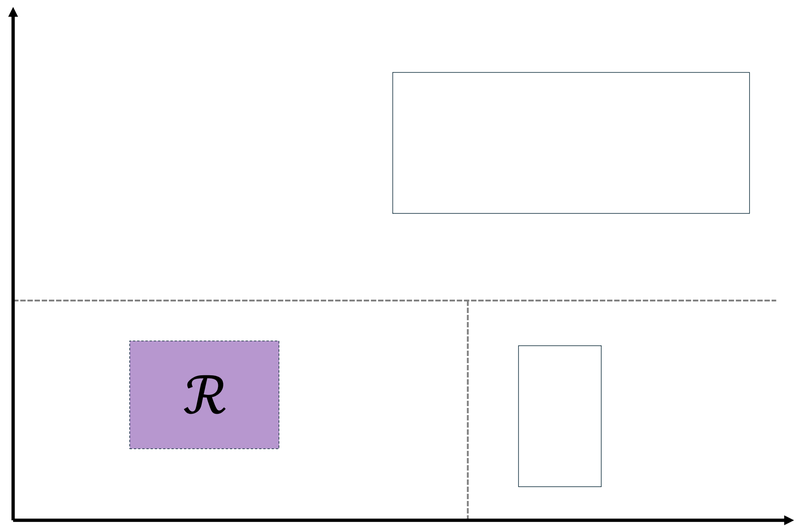
Forgetting procedure



1. Set parent height h as the sum of h_l and h_r,
2. Set parent support $\mathcal{R}$ as the hyperrectangle enclosing $\mathcal{R}_l$ and $\mathcal{R}_r$,

Forgetting procedure





1. Set parent height h as the sum of h_l and h_r,
2. Set parent support $\mathcal{R}$ as the hyperrectangle enclosing $\mathcal{R}_l$ and $\mathcal{R}_r$,
3. Forget split information p, q and child nodes N_l, N_r.

Experimental results

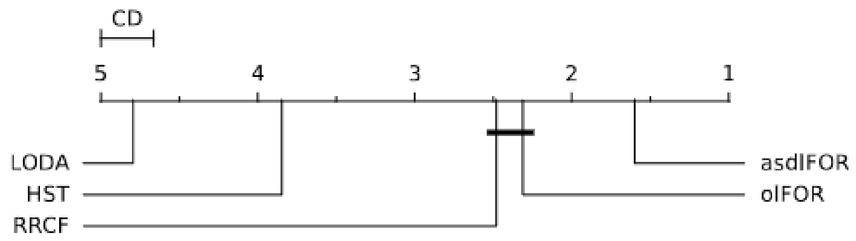
DATASET	n	d	% OF ANOMALIES
DONORS	619326	10	5.90
HTTP	567497	3	0.40
FORESTCOVER	286048	10	0.90
FRAUD	284807	29	0.17
MULCROSS	262144	4	10.00
SMTP	95156	3	0.03
SHUTTLE	49097	9	7.00
MAMMOGRAPHY	11183	6	2.00
NYC_TAXI_SHINGLE	10273	48	5.20
ANNTHYROID	6832	6	7.00
SATELLITE	6435	36	32.00

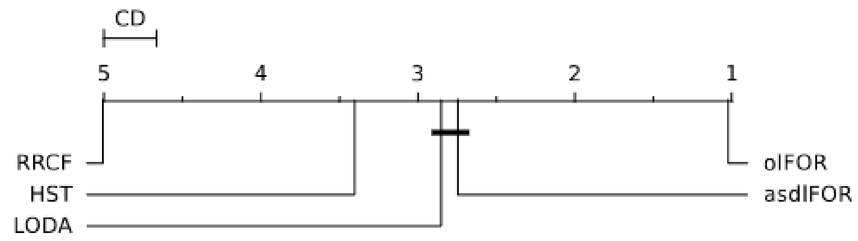
References

- Liu, F. T., Ting, K. M., and Zhou, Z.-H. "Isolation-based anomaly detection". TKDD 2012.
- Ding, Z. and Fei, M. "An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window". IFAC 2013.
- Tan, S. C., Ting, K. M., and Liu, T. F. "Fast anomaly detection for streaming data". JCAI 2011.
- Guha, S., Mishra, N., Roy, G., and Schrijvers, O. "Robust random cut forest based anomaly detection on streams". ICML 2016.
- Pevny, T. "Loda: Lightweight on-line detector of anomalies". Machine Learning 2016.

Experimental results

	AUC (↑)					Time (↓)				
	oIFOR	asdIFOR	HST	RRCF	LODA	oIFOR	asdIFOR	HST	RRCF	LODA
Donors	**0.795**	0.769	0.715	0.637	0.554	**252.36**	551.85	2145.85	4924.46	2111.09
Http	0.998	**0.999**	0.992	0.996	0.632	**179.36**	509.40	2016.00	8367.16	2017.85
ForestCover	0.887	0.861	0.722	**0.917**	0.500	**107.65**	197.82	1045.39	2997.86	1009.92
fraud	0.936	0.946	0.910	**0.951**	0.722	**100.09**	285.69	973.91	4936.03	931.93
Mulcross	**0.995**	0.952	0.011	0.800	0.506	**90.33**	270.79	936.01	3244.96	848.12
Smtp	0.861	**0.905**	0.851	0.894	0.731	**29.95**	142.65	325.77	1273.98	254.69
Shuttle	0.992	**0.996**	0.981	0.957	0.528	**16.35**	108.28	167.48	770.61	130.61
Mammography	0.854	**0.855**	0.831	0.824	0.622	**3.32**	80.01	37.92	118.22	29.55
nyc_taxi_shingle	0.572	0.709	0.342	**0.725**	0.499	**8.03**	83.15	36.70	151.67	36.82
Annthyroid	0.685	**0.810**	0.636	0.740	0.589	**2.00**	77.06	24.26	93.40	19.79
Satellite	0.651	**0.709**	0.531	0.662	0.501	**3.74**	78.90	21.78	93.77	17.55
median	**0.866**	0.863	0.739	0.832	0.541	**29.95**	142.57	323.29	1274.45	254.64
mean rank	2.167	**1.583**	3.917	2.500	4.833	**1.000**	2.667	3.500	5.000	2.833



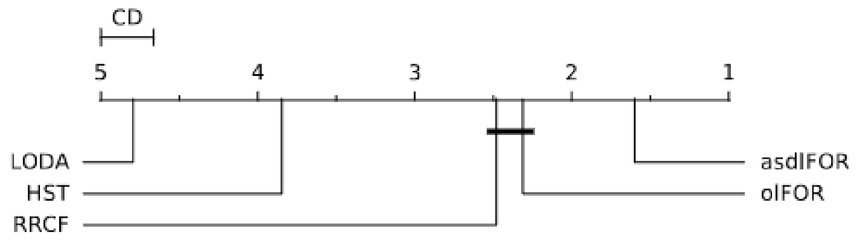


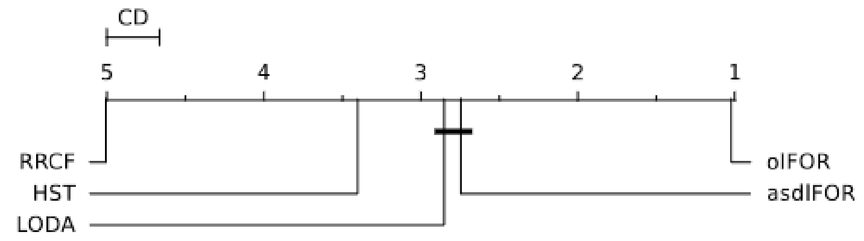
References

- Liu, F. T., Ting, K. M., and Zhou, Z.-H. "Isolation-based anomaly detection". TKDD 2012.
- Ding, Z. and Fei, M. "An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window". IFAC 2013.
- Tan, S. C., Ting, K. M., and Liu, T. F. "Fast anomaly detection for streaming data". JCAI 2011.
- Guha, S., Mishra, N., Roy, G., and Schrijvers, O. "Robust random cut forest based anomaly detection on streams". ICML 2016.
- Pevny, T. "Loda: Lightweight on-line detector of anomalies". Machine Learning 2016.

Experimental results

	AUC (↑)					Time (↓)				
	oIFOR	asdIFOR	HST	RRCF	LODA	oIFOR	asdIFOR	HST	RRCF	LODA
Donors	**0.795**	0.769	0.715	0.637	0.554	**252.36**	551.85	2145.85	4924.46	2111.09
Http	0.998	**0.999**	0.992	0.996	0.632	**179.36**	509.40	2016.00	8367.16	2017.85
ForestCover	0.887	0.861	0.722	**0.917**	0.500	**107.65**	197.82	1045.39	2997.86	1009.92
fraud	0.936	0.946	0.910	**0.951**	0.722	**100.09**	285.69	973.91	4936.03	931.93
Mulcross	**0.995**	0.952	0.011	0.800	0.506	**90.33**	270.79	936.01	3244.96	848.12
Smtp	0.861	**0.905**	0.851	0.894	0.731	**29.95**	142.65	325.77	1273.98	254.69
Shuttle	0.992	**0.996**	0.981	0.957	0.528	**16.35**	108.28	167.48	770.61	130.61
Mammography	0.854	**0.855**	0.831	0.824	0.622	**3.32**	80.01	37.92	118.22	29.55
nyc_taxi_shingle	0.572	0.709	0.342	**0.725**	0.499	**8.03**	83.15	36.70	151.67	36.82
Annthyroid	0.685	**0.810**	0.636	0.740	0.589	**2.00**	77.06	24.26	93.40	19.79
Satellite	0.651	**0.709**	0.531	0.662	0.501	**3.74**	78.90	21.78	93.77	17.55
median	**0.866**	0.863	0.739	0.832	0.541	**29.95**	142.57	323.29	1274.45	254.64
mean rank	2.167	**1.583**	3.917	2.500	4.833	**1.000**	2.667	3.500	5.000	2.833



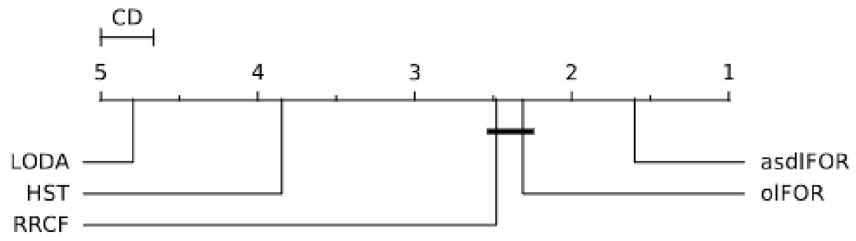


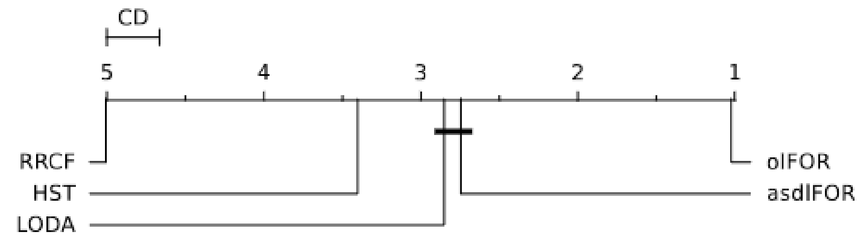
References

- Liu, F. T., Ting, K. M., and Zhou, Z.-H. "Isolation-based anomaly detection". TKDD 2012.
- Ding, Z. and Fei, M. "An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window". IFAC 2013.
- Tan, S. C., Ting, K. M., and Liu, T. F. "Fast anomaly detection for streaming data". JCAI 2011.
- Guha, S., Mishra, N., Roy, G., and Schrijvers, O. "Robust random cut forest based anomaly detection on streams". ICML 2016.
- Pevny, T. "Loda: Lightweight on-line detector of anomalies". Machine Learning 2016.

Experimental results

	AUC (↑)					Time (↓)				
	oIFOR	asdIFOR	HST	RRCF	LODA	oIFOR	asdIFOR	HST	RRCF	LODA
Donors	**0.795**	0.769	0.715	0.637	0.554	**252.36**	551.85	2145.85	4924.46	2111.09
Http	0.998	**0.999**	0.992	0.996	0.632	**179.36**	509.40	2016.00	8367.16	2017.85
ForestCover	0.887	0.861	0.722	**0.917**	0.500	**107.65**	197.82	1045.39	2997.86	1009.92
fraud	0.936	0.946	0.910	**0.951**	0.722	**100.09**	285.69	973.91	4936.03	931.93
Mulcross	**0.995**	0.952	0.011	0.800	0.506	**90.33**	270.79	936.01	3244.96	848.12
Smtp	0.861	**0.905**	0.851	0.894	0.731	**29.95**	142.65	325.77	1273.98	254.69
Shuttle	0.992	**0.996**	0.981	0.957	0.528	**16.35**	108.28	167.48	770.61	130.61
Mammography	0.854	**0.855**	0.831	0.824	0.622	**3.32**	80.01	37.92	118.22	29.55
nyc_taxi_shingle	0.572	0.709	0.342	**0.725**	0.499	**8.03**	83.15	36.70	151.67	36.82
Annthyroid	0.685	**0.810**	0.636	0.740	0.589	**2.00**	77.06	24.26	93.40	19.79
Satellite	0.651	**0.709**	0.531	0.662	0.501	**3.74**	78.90	21.78	93.77	17.55
median	**0.866**	0.863	0.739	0.832	0.541	**29.95**	142.57	323.29	1274.45	254.64
mean rank	2.167	**1.583**	3.917	2.500	4.833	**1.000**	2.667	3.500	5.000	2.833



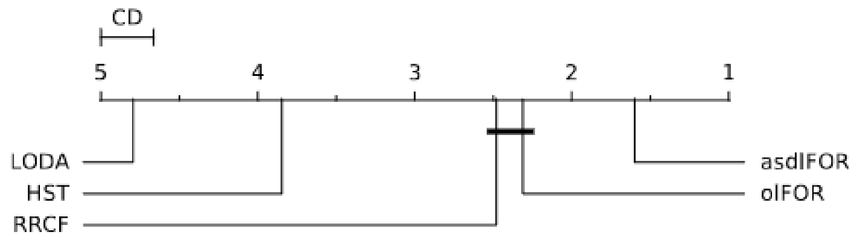


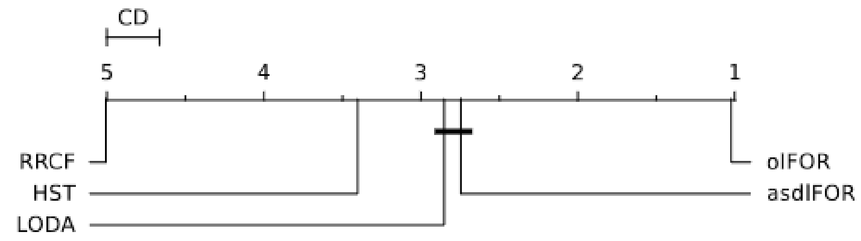
References

- Liu, F. T., Ting, K. M., and Zhou, Z.-H. "Isolation-based anomaly detection". TKDD 2012.
- Ding, Z. and Fei, M. "An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window". IFAC 2013.
- Tan, S. C., Ting, K. M., and Liu, T. F. "Fast anomaly detection for streaming data". JCAI 2011.
- Guha, S., Mishra, N., Roy, G., and Schrijvers, O. "Robust random cut forest based anomaly detection on streams". ICML 2016.
- Pevny, T. "Loda: Lightweight on-line detector of anomalies". Machine Learning 2016.

Experimental results

	AUC (↑)					Time (↓)				
	oIFOR	asdIFOR	HST	RRCF	LODA	oIFOR	asdIFOR	HST	RRCF	LODA
Donors	**0.795**	0.769	0.715	0.637	0.554	**252.36**	551.85	2145.85	4924.46	2111.09
Http	0.998	**0.999**	0.992	0.996	0.632	**179.36**	509.40	2016.00	8367.16	2017.85
ForestCover	0.887	0.861	0.722	**0.917**	0.500	**107.65**	197.82	1045.39	2997.86	1009.92
fraud	0.936	0.946	0.910	**0.951**	0.722	**100.09**	285.69	973.91	4936.03	931.93
Mulcross	**0.995**	0.952	0.011	0.800	0.506	**90.33**	270.79	936.01	3244.96	848.12
Smtp	0.861	**0.905**	0.851	0.894	0.731	**29.95**	142.65	325.77	1273.98	254.69
Shuttle	0.992	**0.996**	0.981	0.957	0.528	**16.35**	108.28	167.48	770.61	130.61
Mammography	0.854	**0.855**	0.831	0.824	0.622	**3.32**	80.01	37.92	118.22	29.55
nyc_taxi_shingle	0.572	0.709	0.342	**0.725**	0.499	**8.03**	83.15	36.70	151.67	36.82
Annthyroid	0.685	**0.810**	0.636	0.740	0.589	**2.00**	77.06	24.26	93.40	19.79
Satellite	0.651	**0.709**	0.531	0.662	0.501	**3.74**	78.90	21.78	93.77	17.55
median	**0.866**	0.863	0.739	0.832	0.541	**29.95**	142.57	323.29	1274.45	254.64
mean rank	2.167	**1.583**	3.917	2.500	4.833	**1.000**	2.667	3.500	5.000	2.833





References

- Liu, F. T., Ting, K. M., and Zhou, Z.-H. "Isolation-based anomaly detection". TKDD 2012.
- Ding, Z. and Fei, M. "An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window". IFAC 2013.
- Tan, S. C., Ting, K. M., and Liu, T. F. "Fast anomaly detection for streaming data". JCAI 2011.
- Guha, S., Mishra, N., Roy, G., and Schrijvers, O. "Robust random cut forest based anomaly detection on streams". ICML 2016.
- Pevny, T. "Loda: Lightweight on-line detector of anomalies". Machine Learning 2016.

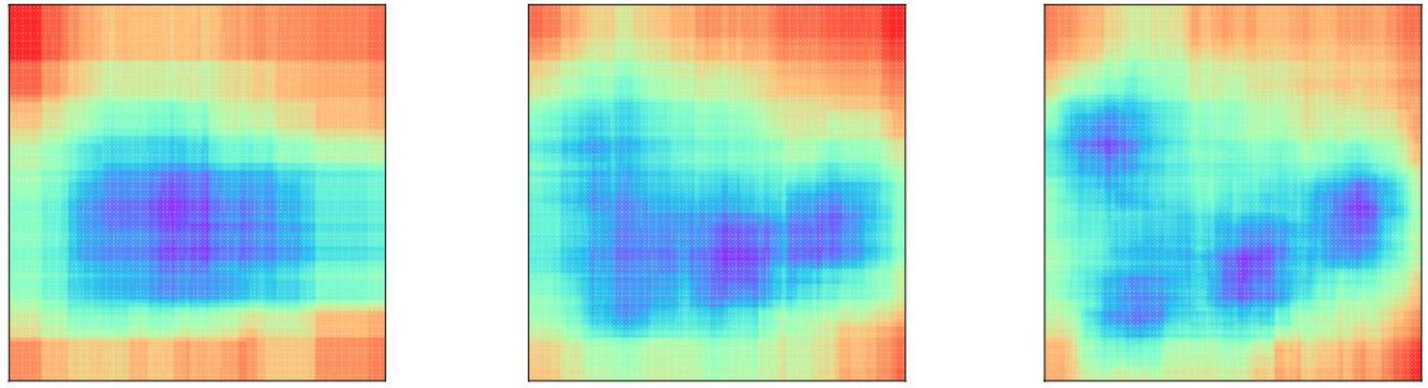
Conclusions 

Online Isolation Forest is a truly online anomaly detection algorithm that **tracks the data generating process** as it evolves over time.

Experimental validation demonstrated that Online Isolation Forest:

- is **on par** with state-of-the art techniques in terms of **effectiveness**,
- consistently **outperforms** all competitors in terms of **efficiency**.

Thank you



PAPER

PROJECT PAGE

filippo.leveni@polimi.it

POLITECNICO MILANO 1863

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

FUTURE AI RESEARCH