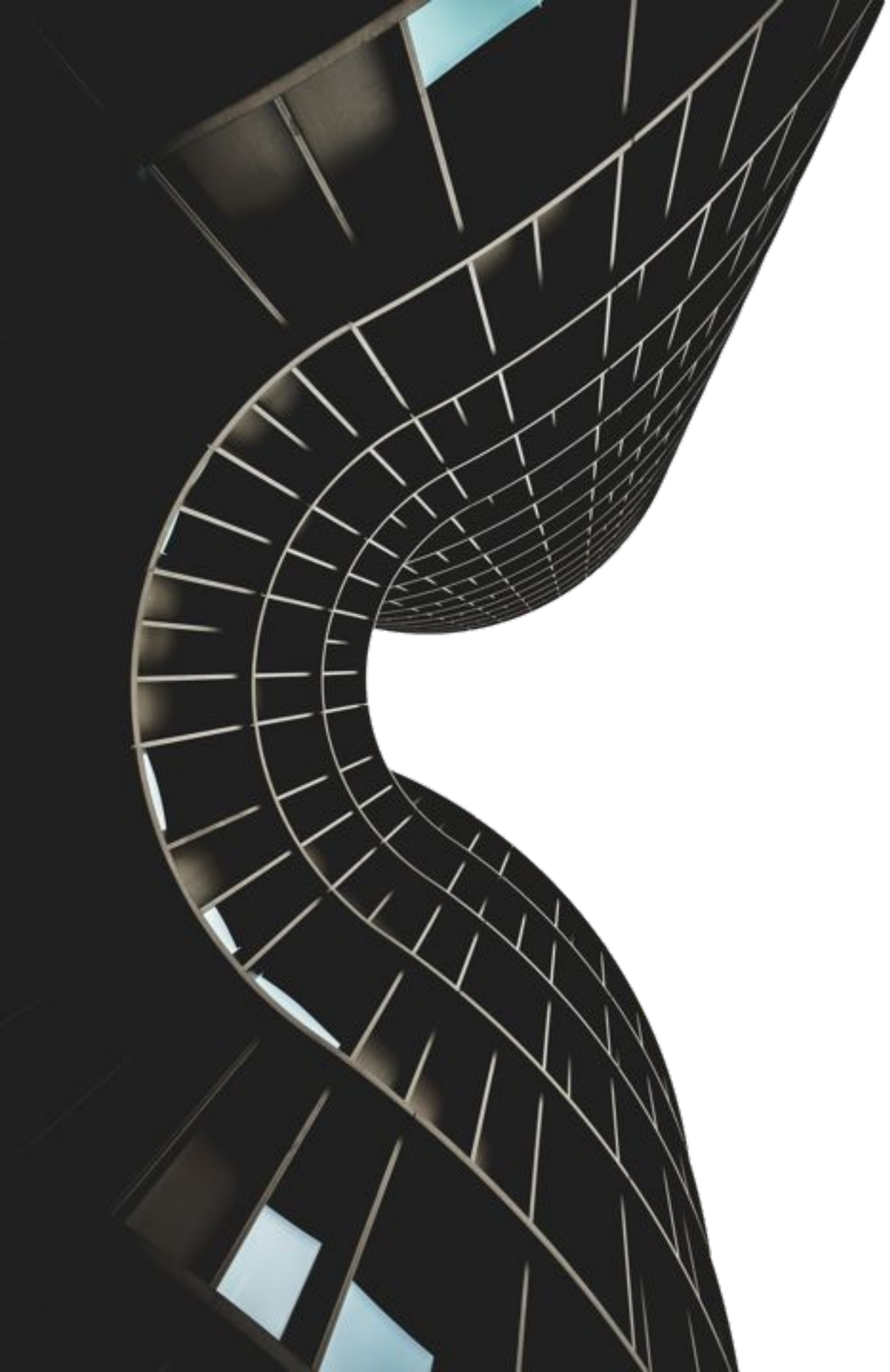


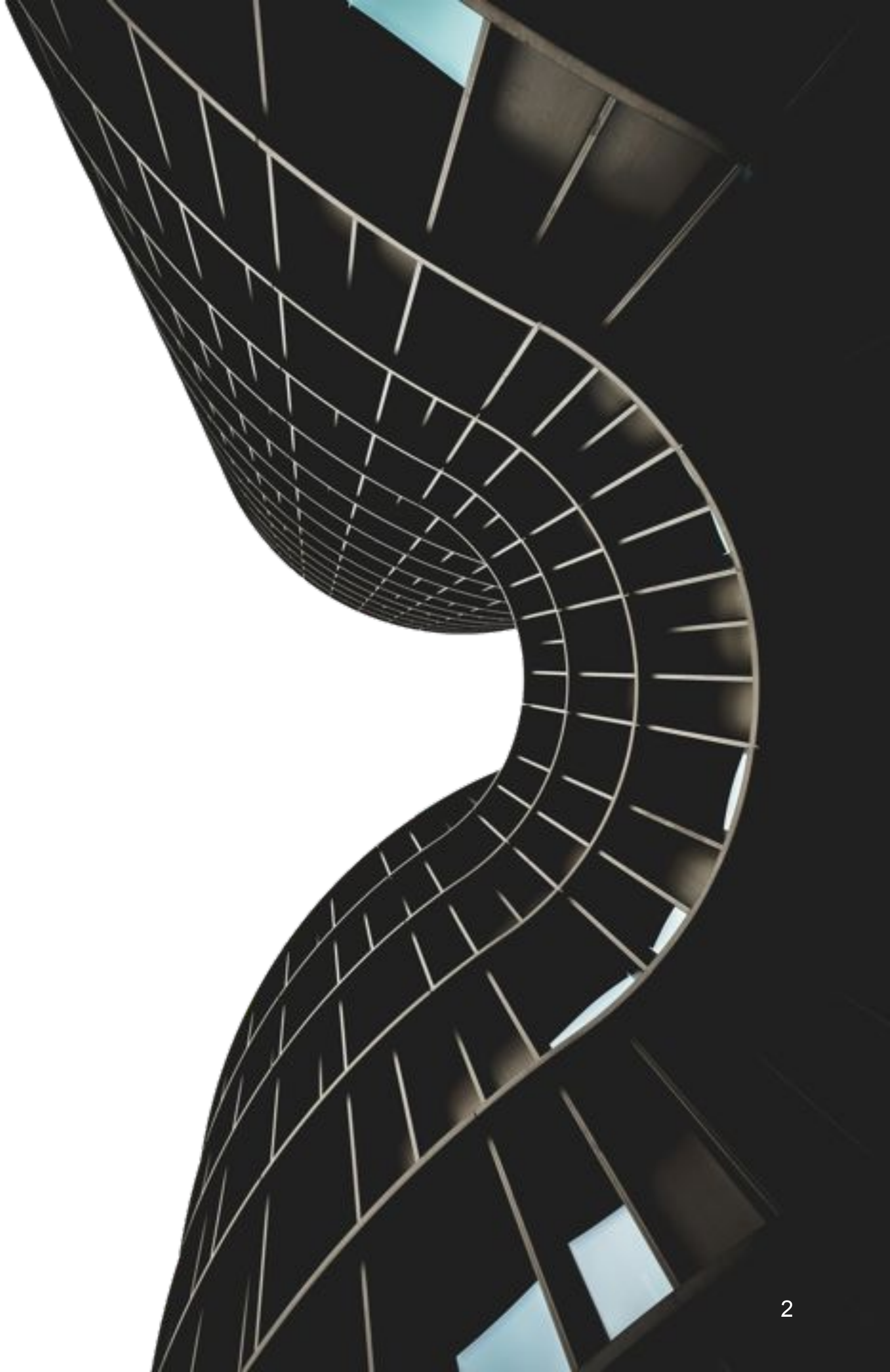
feedzai

Lightweight Real-Time Feature Monitoring

Master Thesis Defense

July 24th, 2020

Context

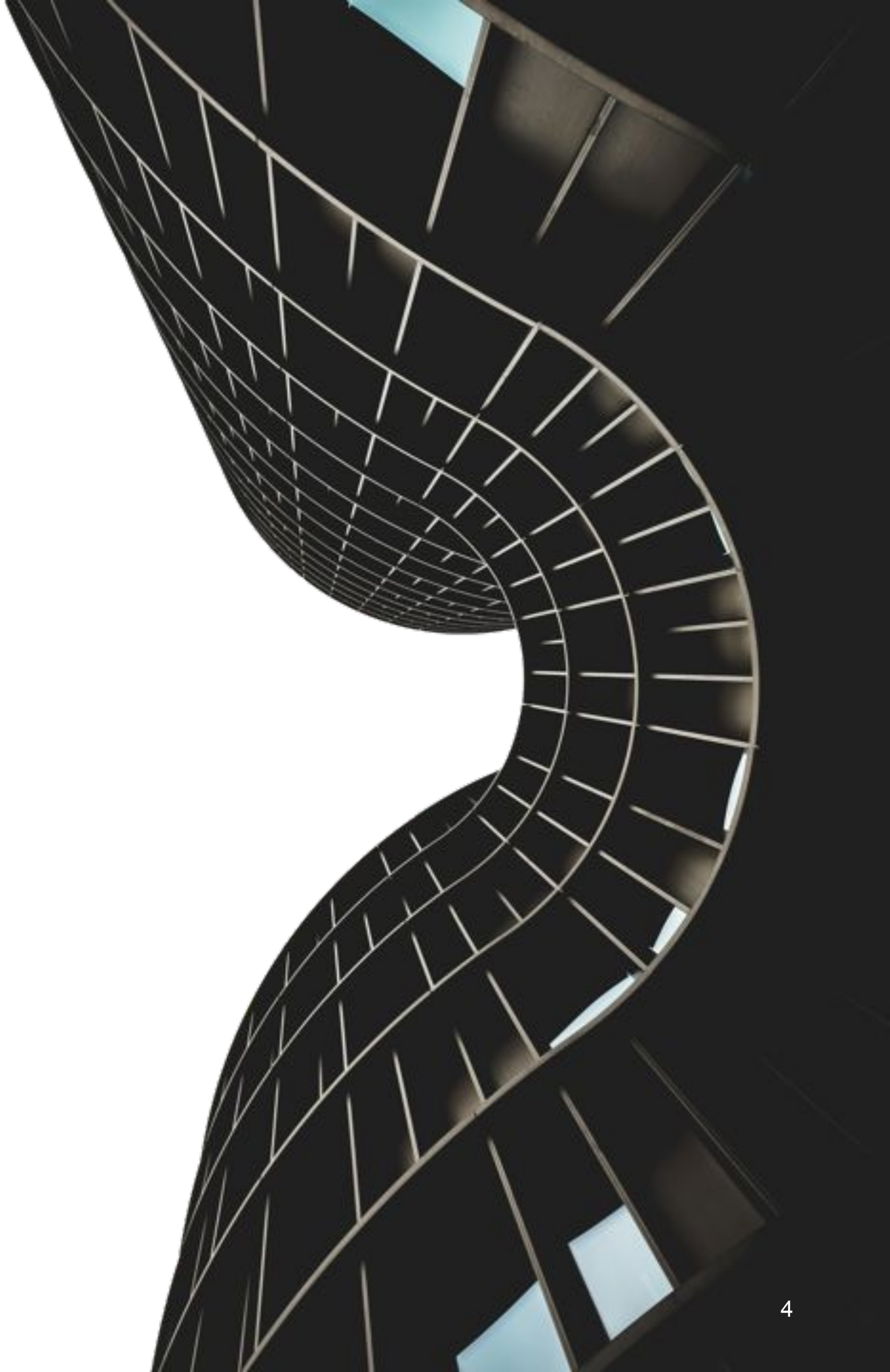


Context

Feedzai working context

- Feedzai uses **Machine Learning (ML)** models to monitor streams of credit card transactions to **detect fraud**.
- In the field of ML, change in distributions of the data stream is known as **concept drift** [1,2].
- In the presence of concept drift, the **performance** of these models **gradually deteriorates**.
- Feedzai has model monitoring systems that alert for this **performance decay**, indicating the **model needs retraining**.

Challenge



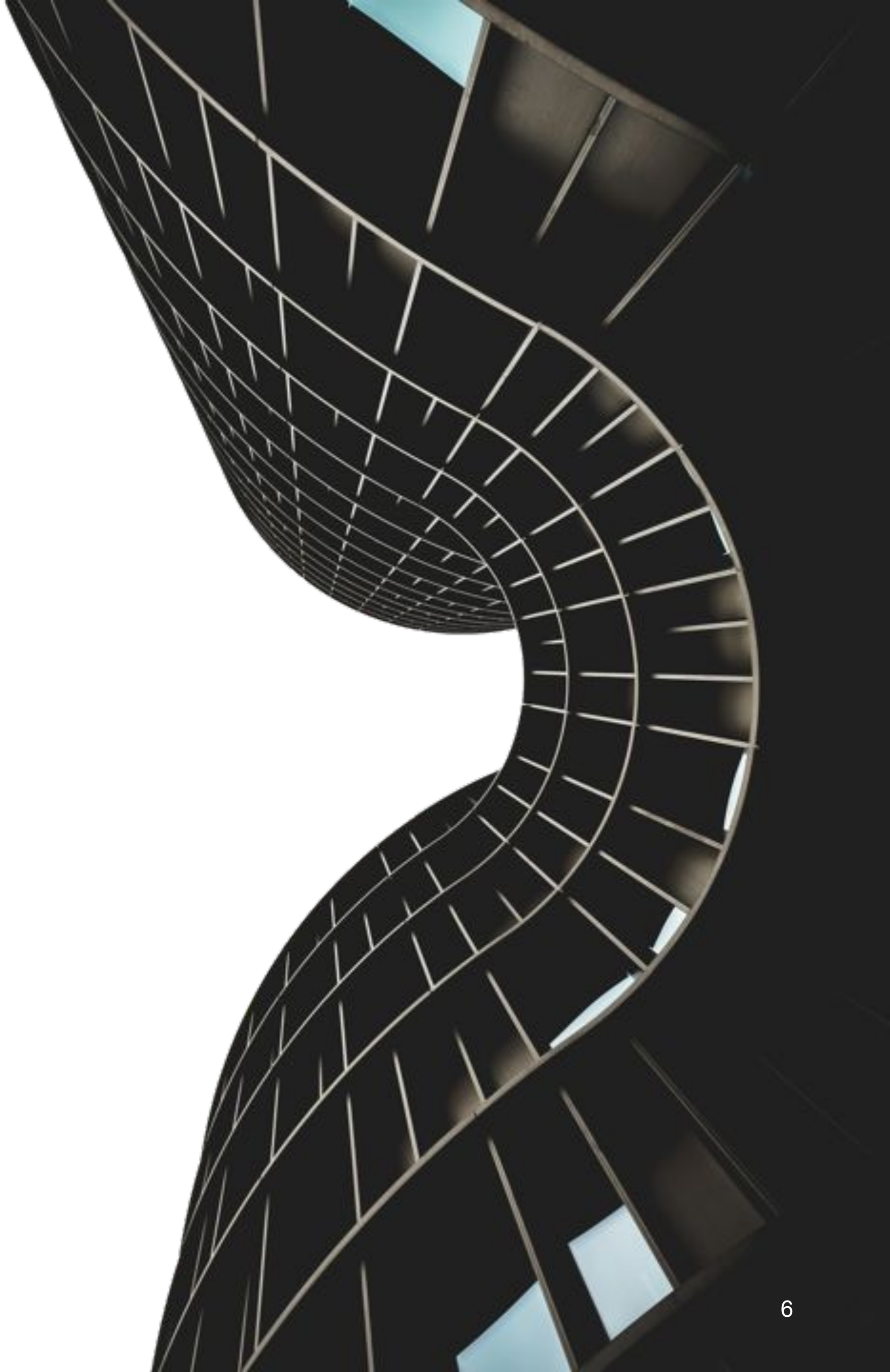
Challenge

- **Detect concept drift and alert** system administrators so they can **reconfigure** their **systems before** its **performance declines**.

- Move from a **"Train, Detect performance decay and Retrain"** paradigm to a **"Train, Detect concept drift and Retrain"** paradigm.

- Devise a **generic method** that **assists any system** (not just ML models) that **learn from past data** and whose **performance depends on** the how well the **assumptions made** hold for future data.

Previous Work



Previous Work

- **SAMM** [3] is an unsupervised **concept drift detection** method for data streams. Identifies **short term abrupt changes.**

- SAMM **monitors** the prediction **scores of ML models**.

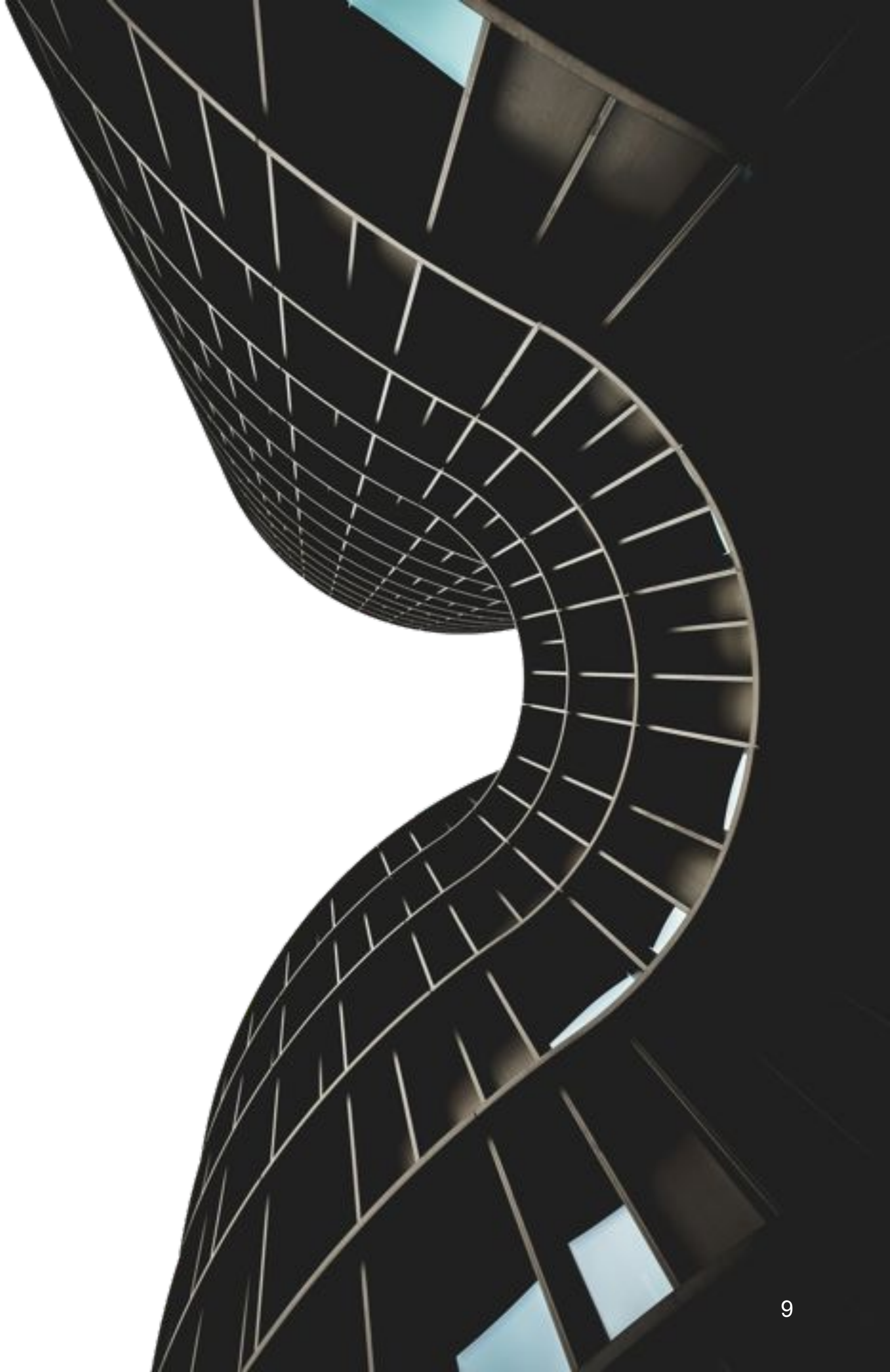
Issue #1: does so based on model scores, so the changes already **impacted the model** (our system **performance already suffered**).

Previous Work

- SAMM works with a **two-windowed model**: a static one and a sliding one over the data stream.

  Issue #2: it keeps both in memory **(linear memory complexity)**, so it is used with small windows.

Objectives



Objectives

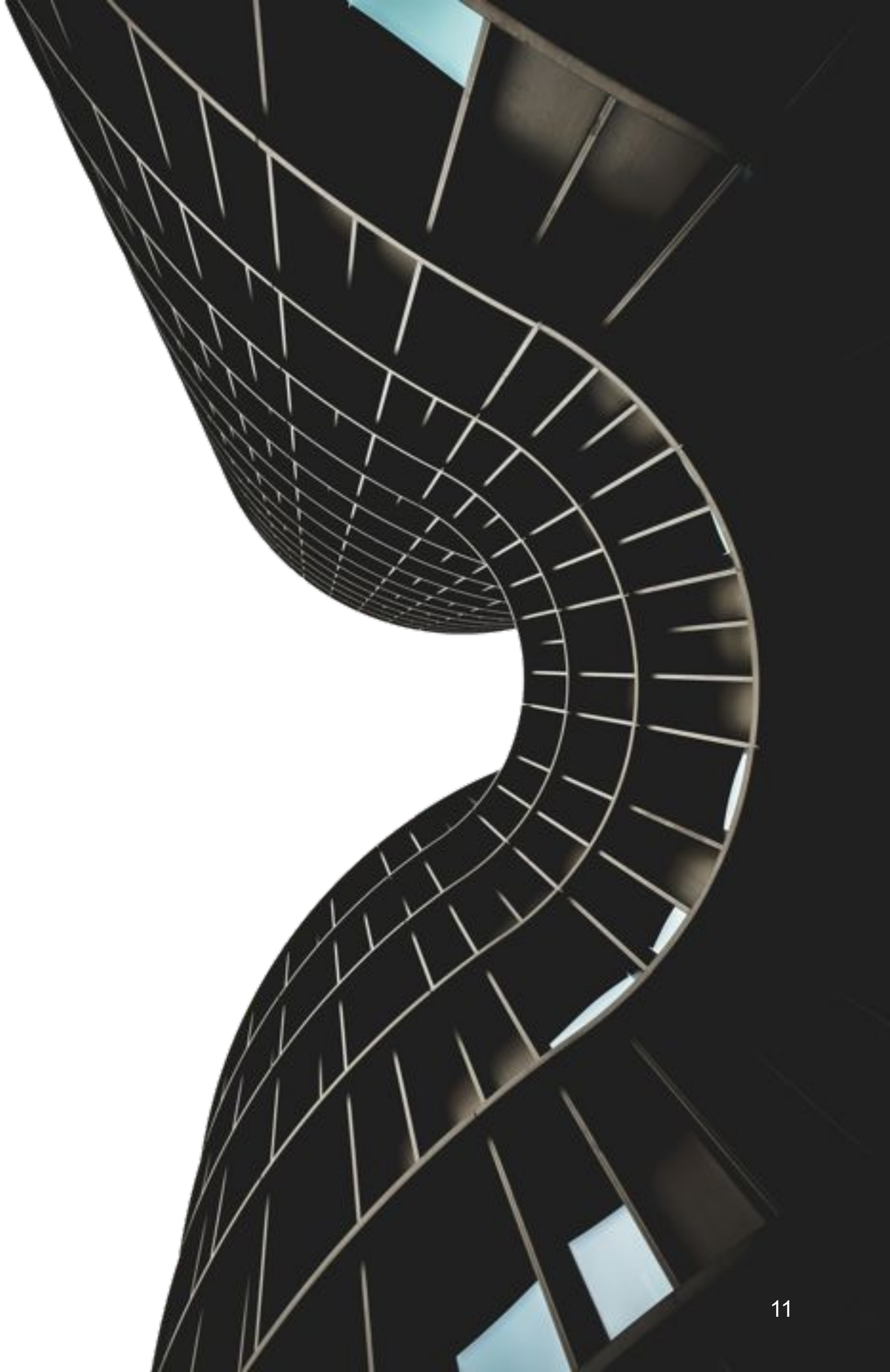
- **Lightweight**: memory complexity grows sublinearly with window size.
- **Real-time:** process data stream events in real-time.
- **Feature Monitoring:** monitor data patterns and **alert for changes** in the underlying distribution of **individual features**.

Detect **data pattern shifts** and **alert admins of** a third party **mission-critical system** that performs an analysis on the stream of data and **depends on assumptions made** over a fixed reference period **before its performance decays.**

feedzai

Feature Monitoring

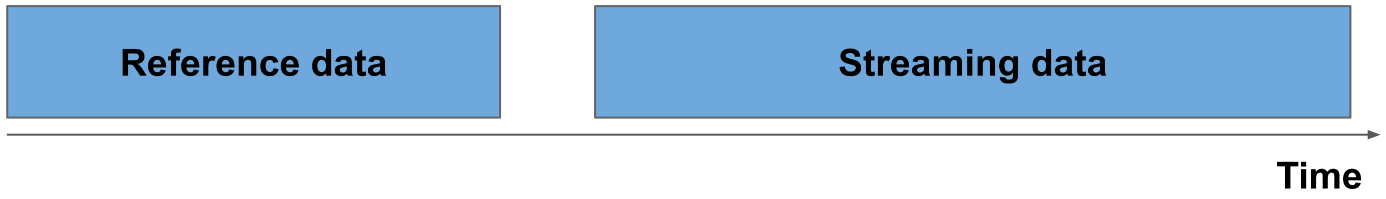
A two-phased method



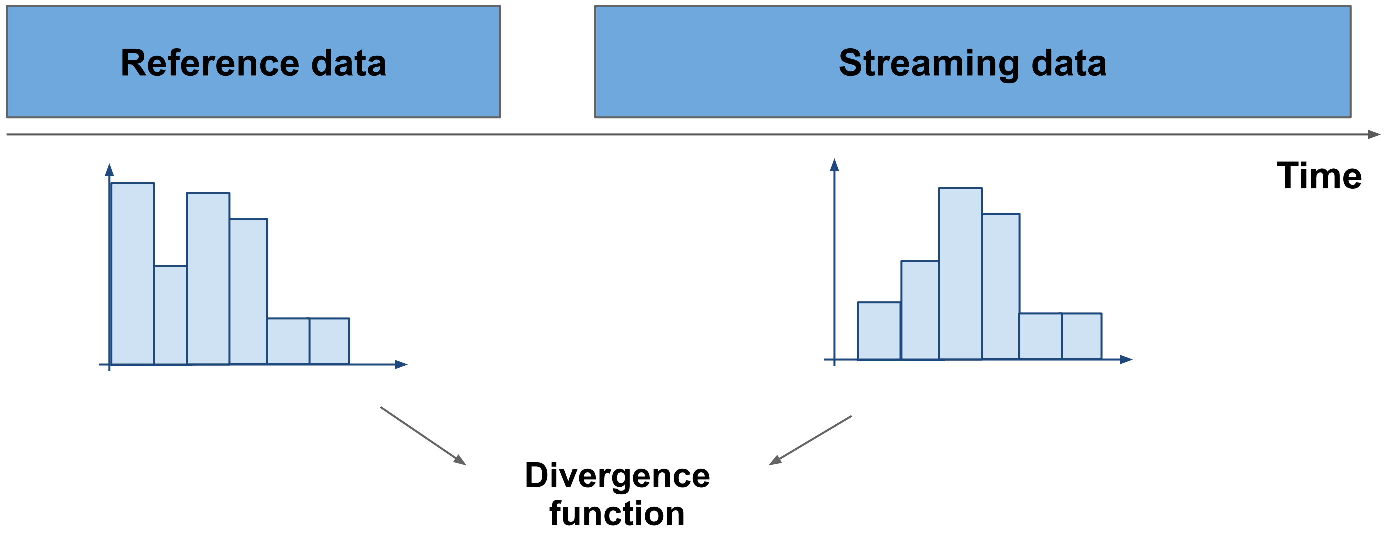
A two-phased method

- Our method has two phases:
  1. A **batch analysis** over the **reference period**.
  2. A **stream analysis** over the **target period** (online event-by-event processing).
- The **reference window** is static: it is the **ground-truth** of our system from which we make our assumptions (e.g., the data used for model training).
- The **target window** is a simulated sliding window of **streaming data** (e.g., transactions a model sees in production).
- Our method works for **numerical (and encoded categoricals) features** only.

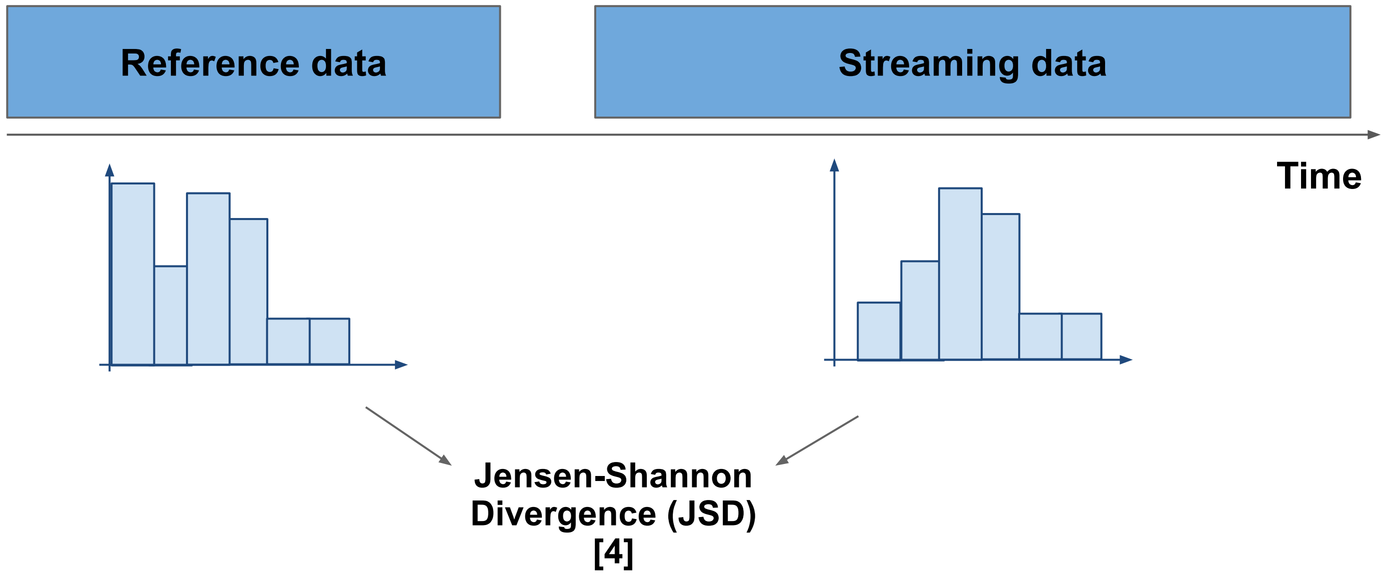
An overview



An overview

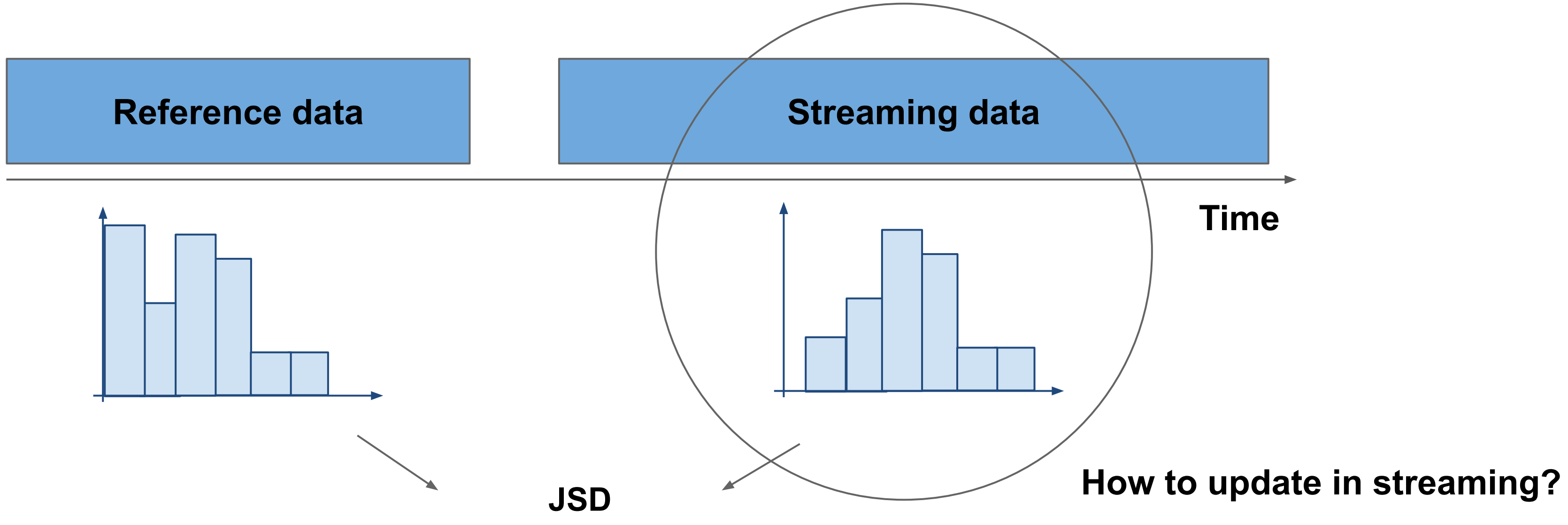


An overview



Reference data
Streaming data
Time
Jensen-Shannon
Divergence (JSD)
[4]

An overview

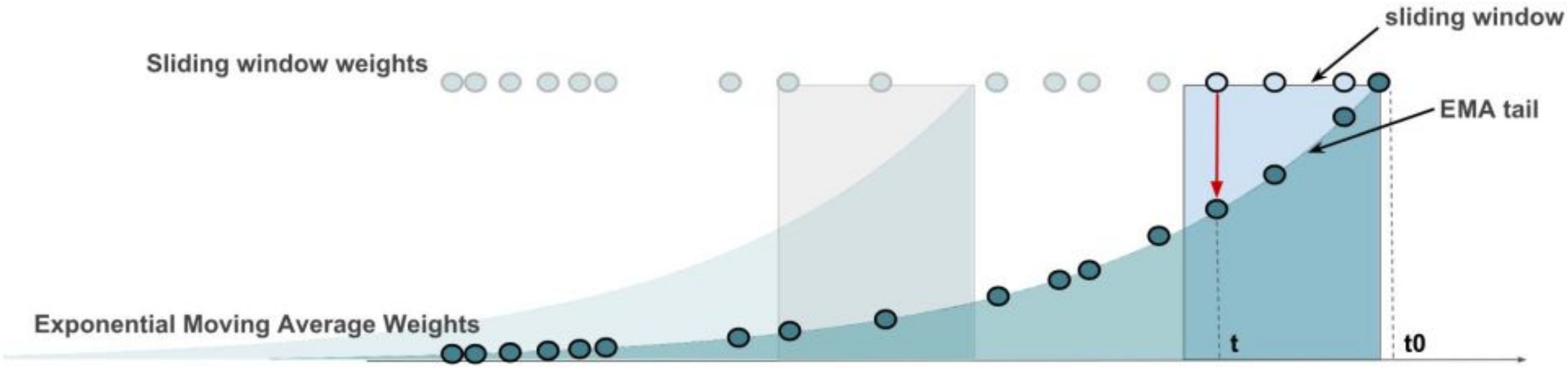

Reference data
Streaming data
Time
JSD
How to update in streaming?

Streaming Histogram

- The streaming histogram has to be updated.
- That implies **inserting new events and expiring old ones**.
- The **“subtract-on-evict” method has linear memory complexity** regarding the sliding window size, as we have to keep every value in memory to know when to remove the oldest.
- We propose an **approximated histogram aggregation based on Exponential Moving Averages (EMAs)**.

Exponential Moving Averages

- An **Exponential Moving Average (EMA)** [5,6,7] is a type of moving average that places a **greater weight and significance on the most recent data points**.



Exponential Moving Averages

EMA-count based histogram

- EMAs are **recursive:**

$$\mathbf{EMA_t = val_t + EMA_{t-1} * smoothing\_factor}$$

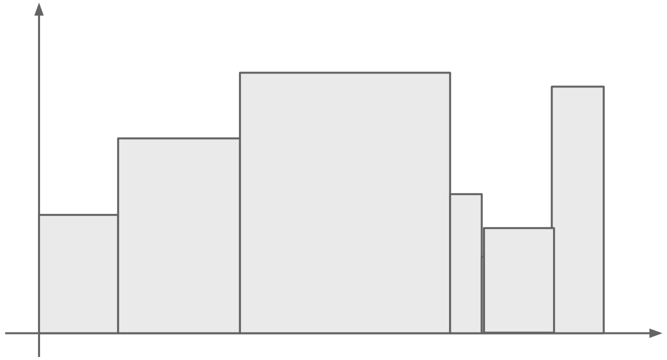
- The approximated histogram is made up of N bins and **each bin holds an EMA-count.**

- **Constant memory complexity** (number of bins does not grow).

- Updated in **constant time** (simple increment times suppression).

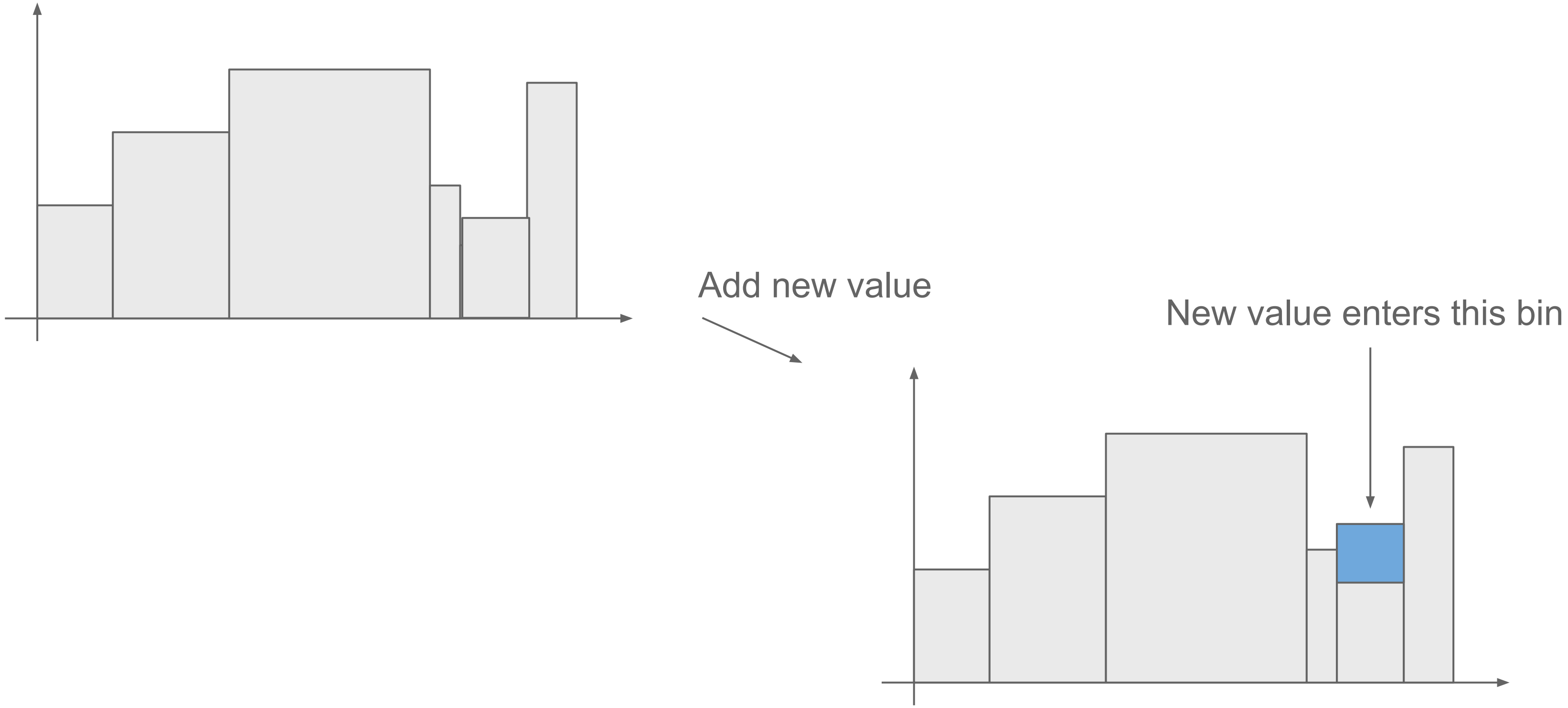
Approximated histogram aggregation

EMA-like histogram



Approximated histogram aggregation

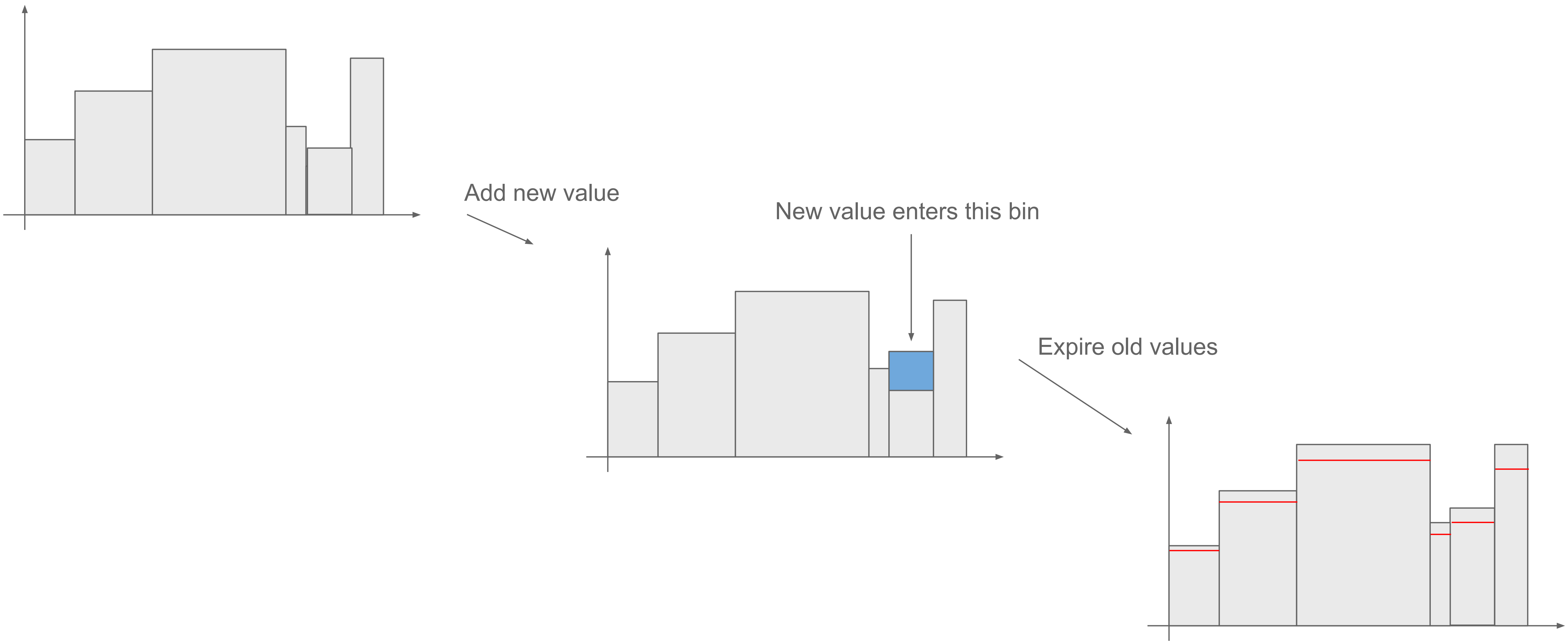
EMA-like histogram



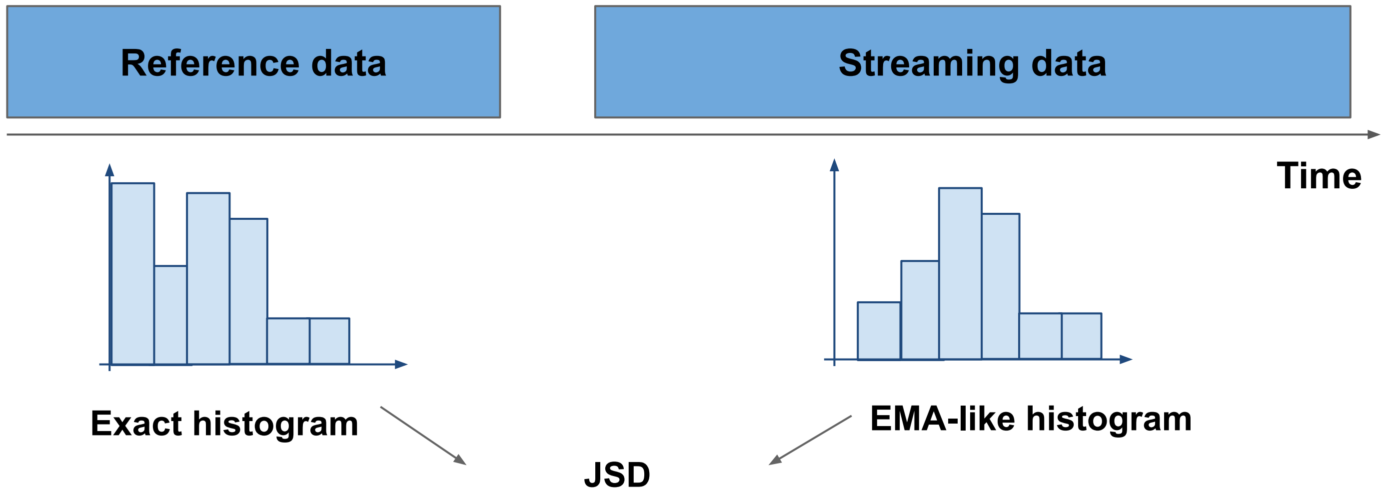
feedzai

Approximated histogram aggregation

EMA-like histogram



An overview


Reference data
Streaming data
Time
Exact histogram
EMA-like histogram
JSD

Batch analysis

A two-phased method

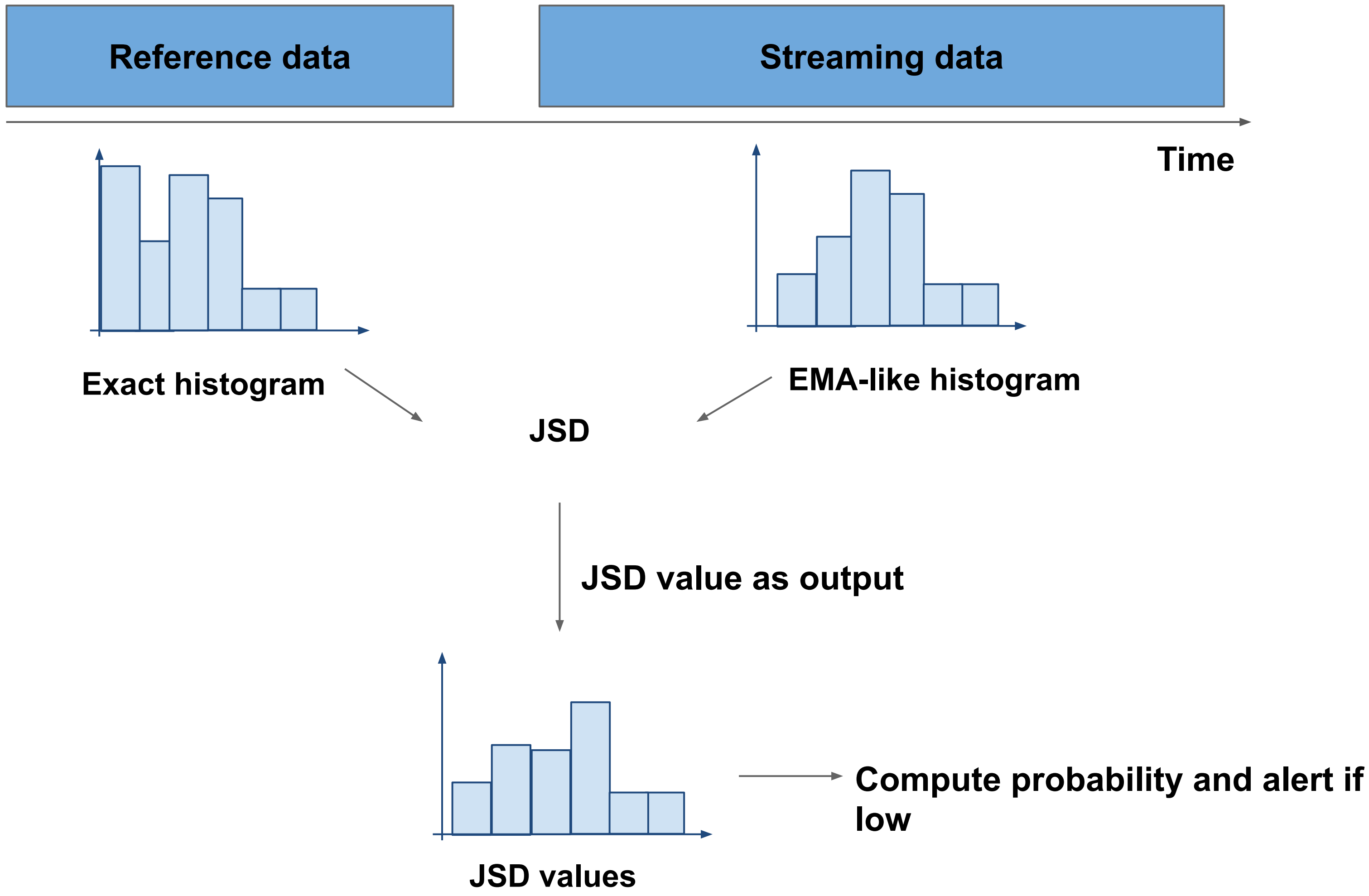
- But what **JSD value should we expect**?

- Somehow we need to **encode the probability of this JSD value** to raise an alert if low.

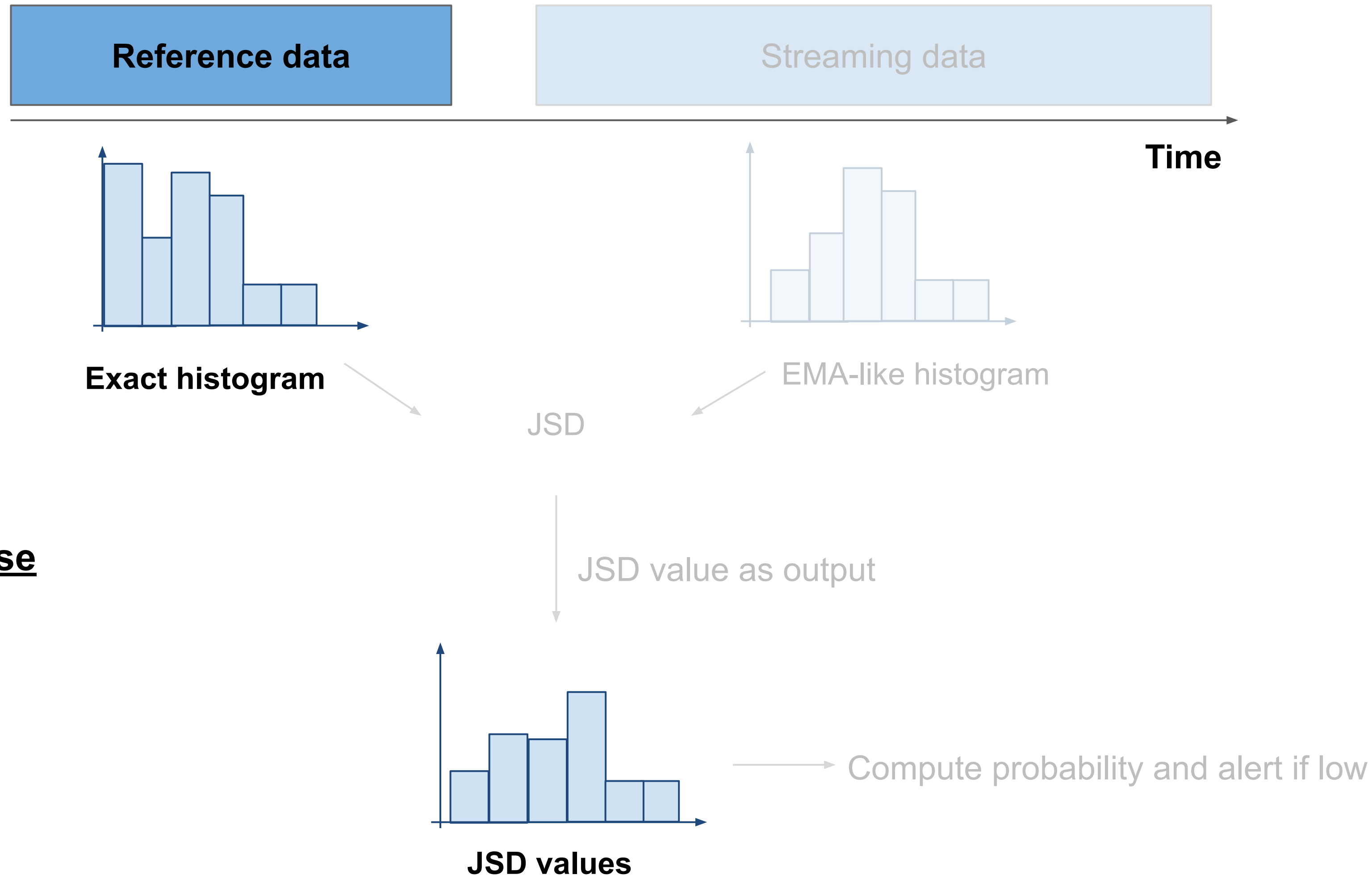
- To achieve this we build a **distribution of JSD values** through a sampling process in batch.

feedzai

An overview

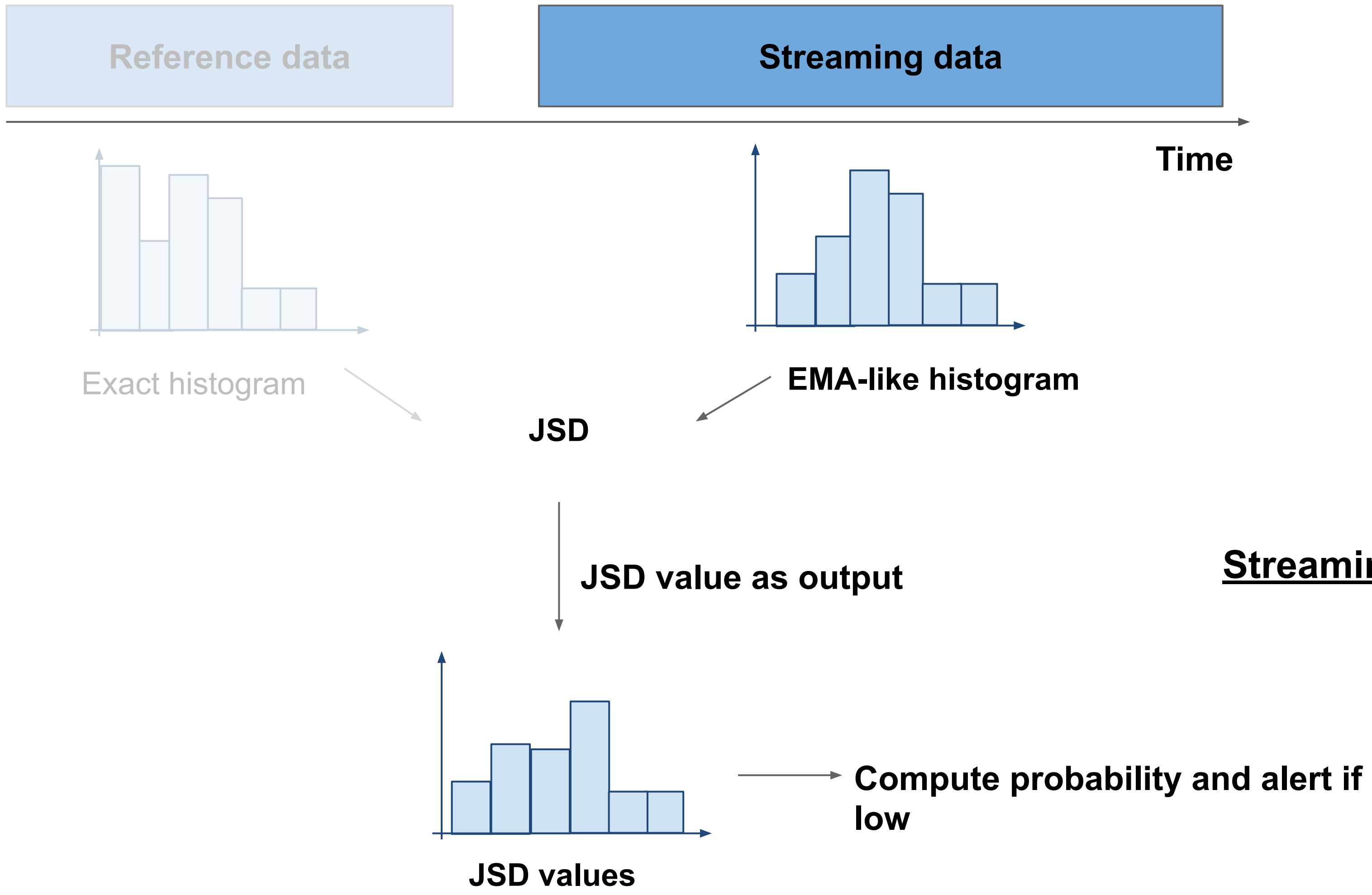


An overview



<u>Batch Phase</u>

An overview

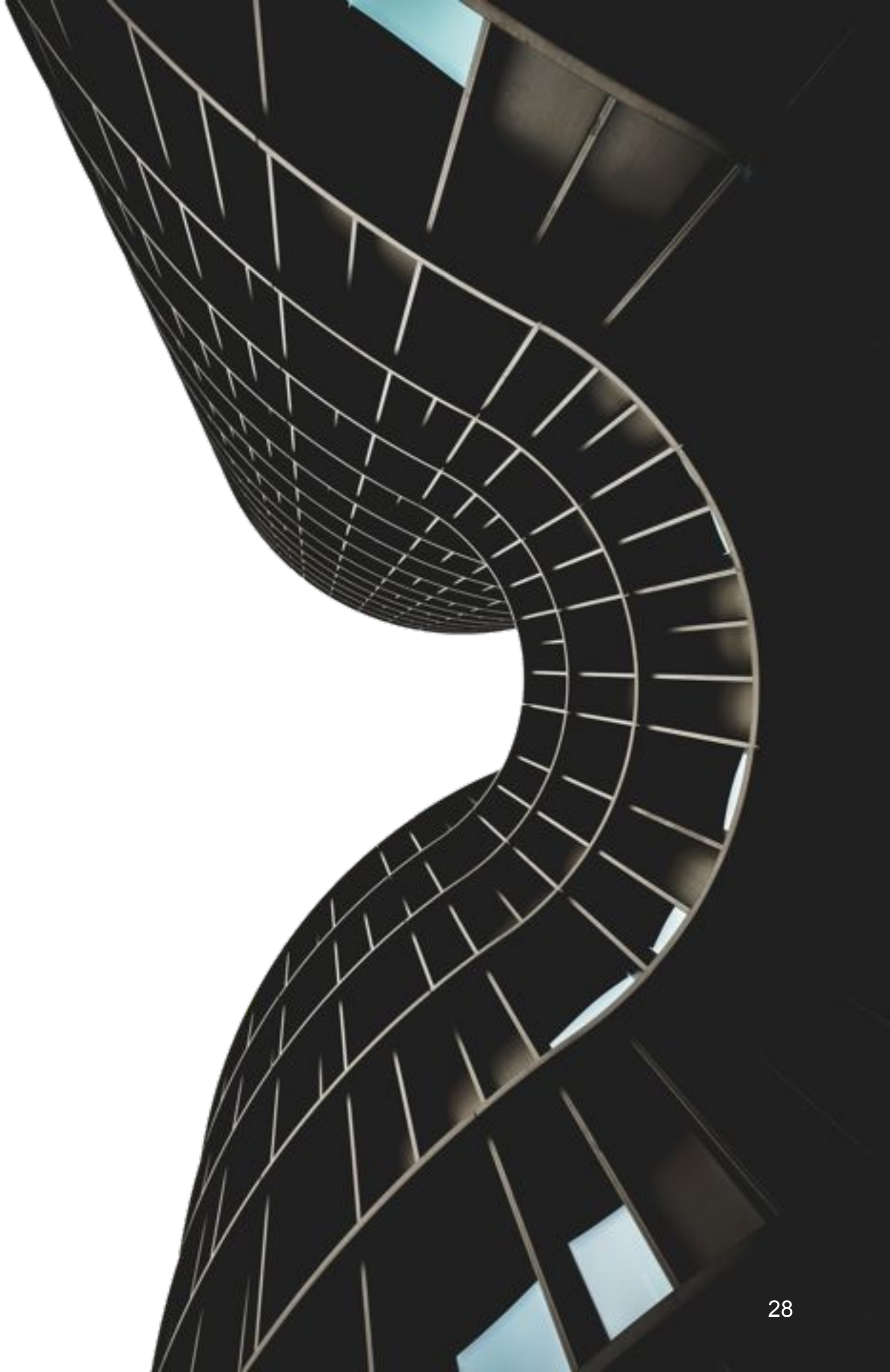


Streaming Phase

feedzai

Batch Phase

A two-phased method

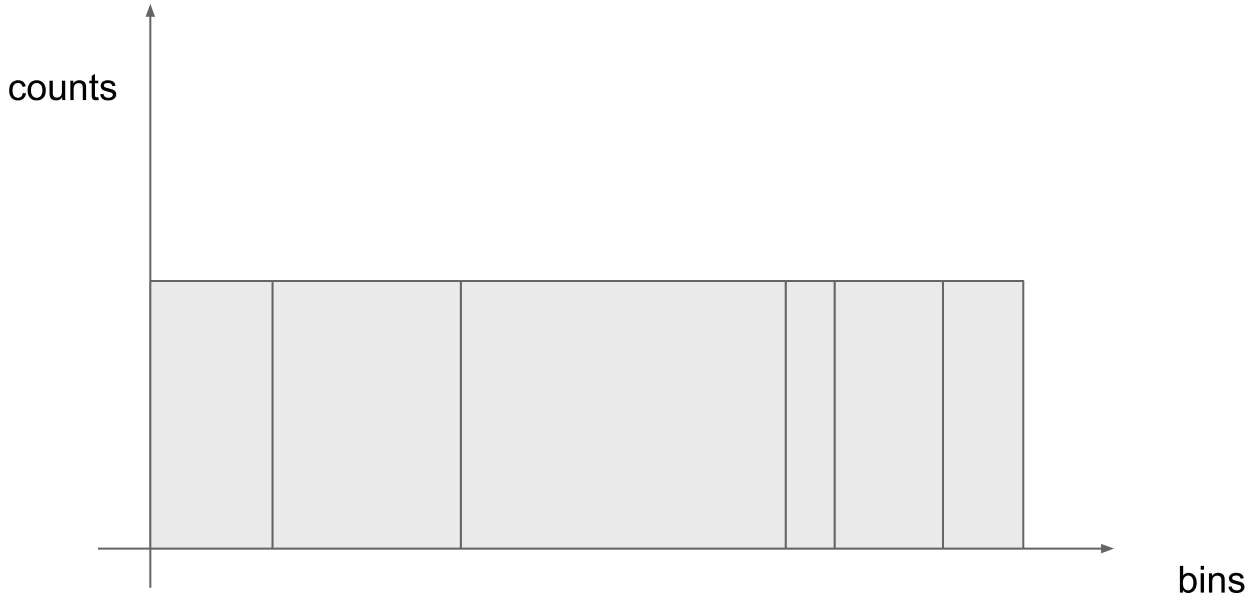


Batch analysis

A two-phased method

For each feature:

1. Compute the reference histogram from the reference data set having **equal counts per bin** (implies different sized bins).

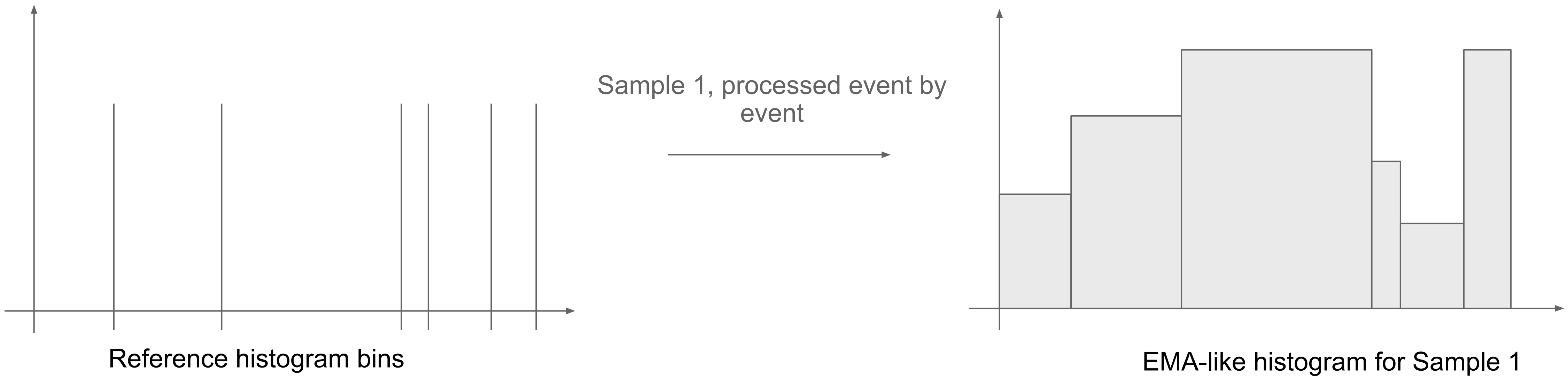


Batch analysis

A two-phased method

2. Make **S samples** of transactions, each with K tuples.

   For each sample, compute an approximated **EMA histogram** using the **bins** from the **reference histogram**.

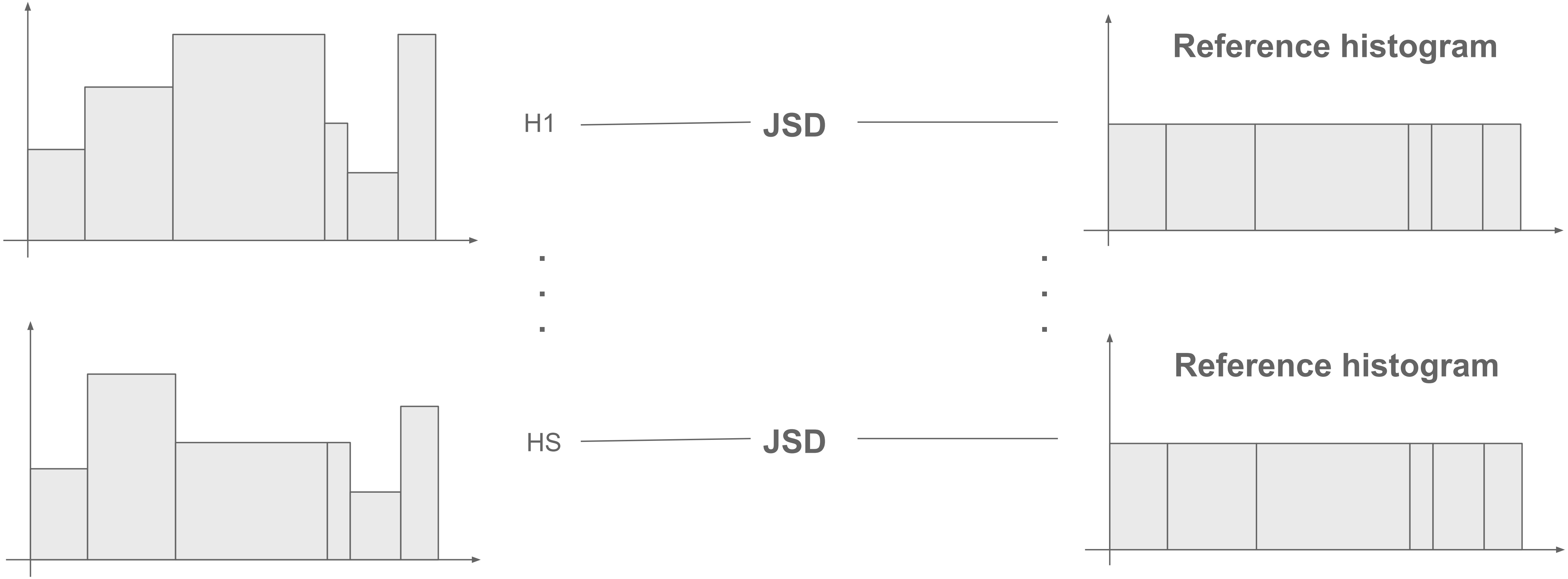


feedzai

Batch analysis

A two-phased method

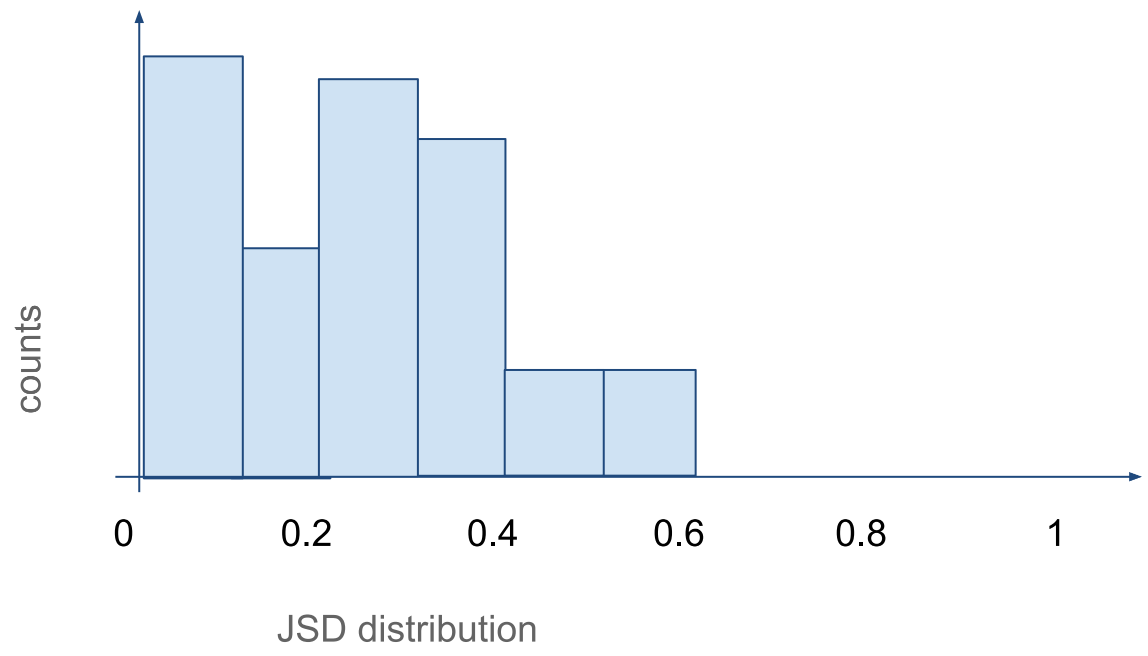
3. Compute **S JSDs** between the **S histograms and the reference one**.



Batch analysis

A two-phased method

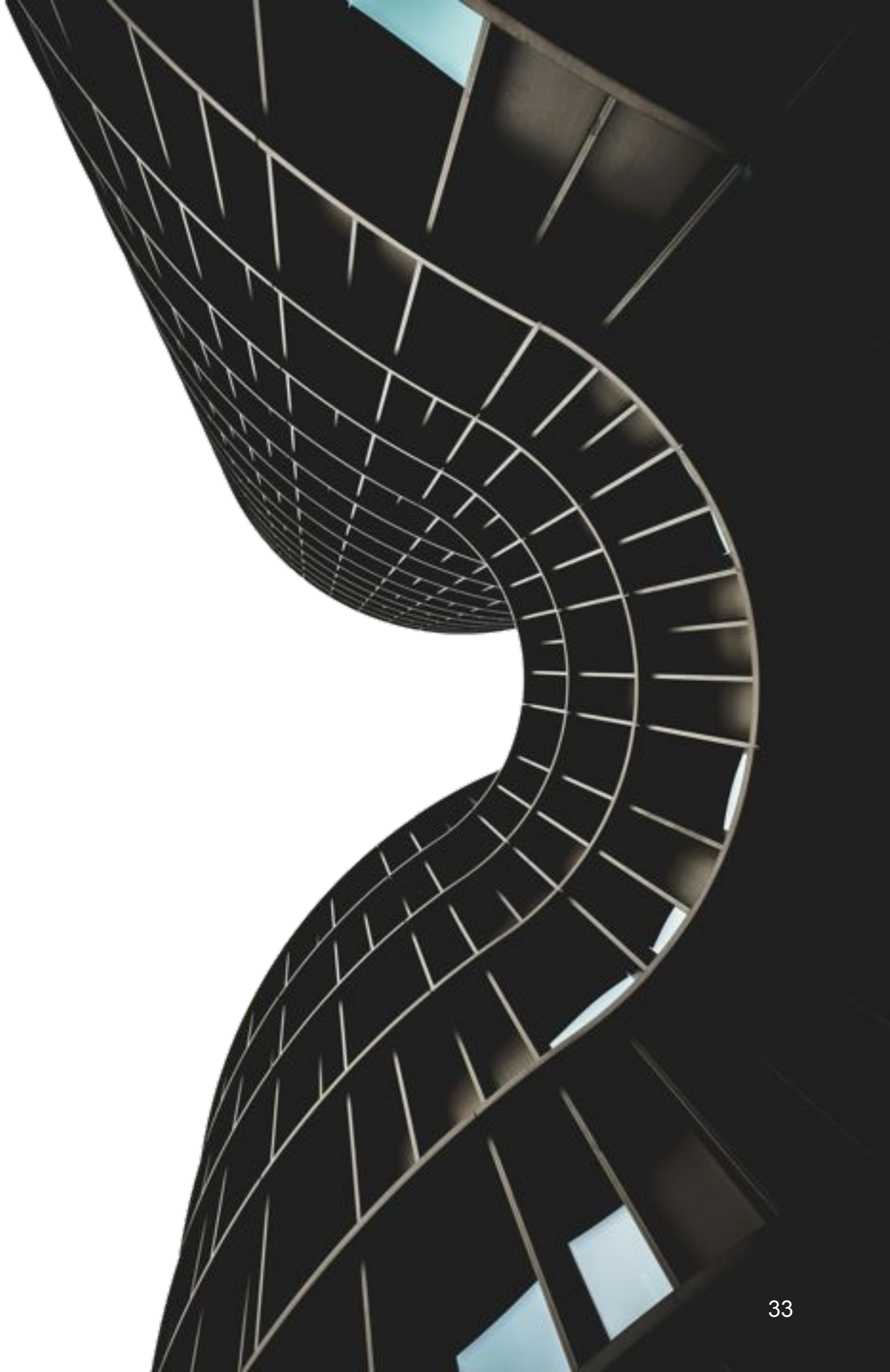
- Build the histogram below that **encodes the distribution of JSD values** using the S **JSD values from the sampling process**.



feedzai

Streaming Phase

A two-phased method



Stream analysis

A two-phased method

For each feature:

- Initialize the **EMA histogram** as the last sample's histogram from batch analysis to simulate the burn-in period.

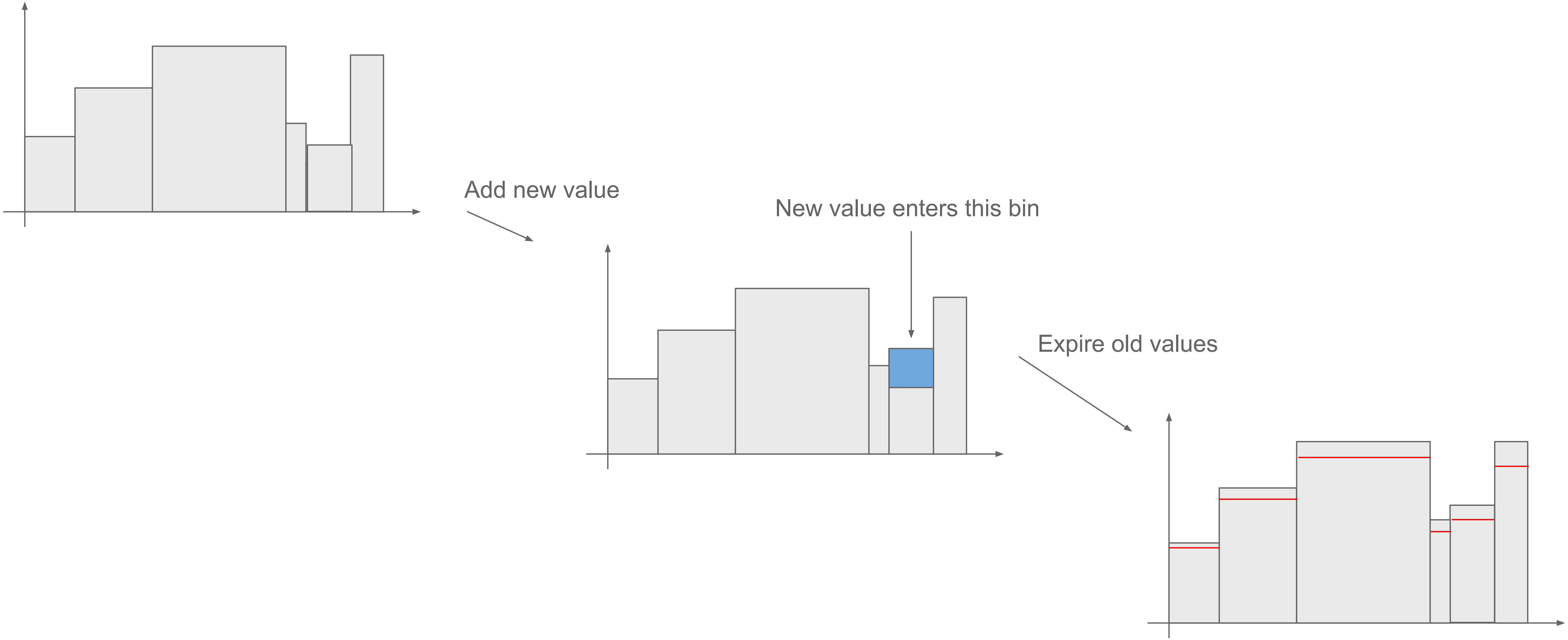
For each event and for each feature:

1. Update the **EMA histogram**.

   Add the value to the correct bin and then apply the suppression factor to all bins.

Approximated histogram aggregation

Recall



Stream analysis

A two-phased method

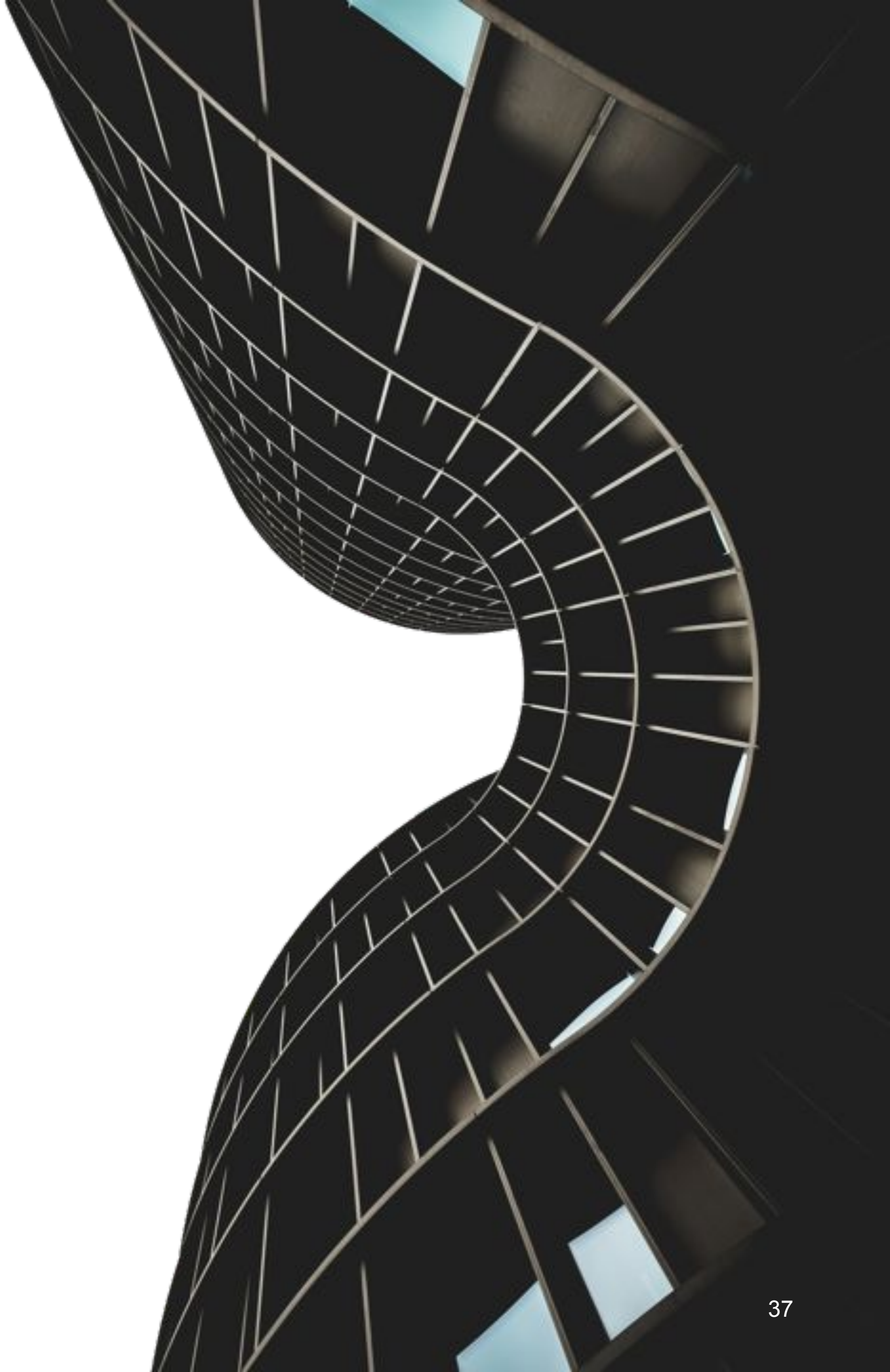
Periodically apply the divergence test:

2. Compute **JSD between the reference histogram and the streaming histogram**.

3. Test in which percentile the **JSD value** is in the histogram of **JSDs of the batch analysis** and alert values with low probabilities (low p-values).

Divergence Test

The multiple comparison problem



Multiple Testing Correction

Online phase alerts

- **For each feature** we compute a **JSD value**.
- **For each JSD value** we compute its **probability** and **check if it is lower than** a user predefined **threshold**.
- When repeating this analysis for multiple features we encounter a **multiple comparison problem** [8,9,10,11].
- **The more comparisons made, the larger the false positive rate**: considering one hypothesis true when it is not.

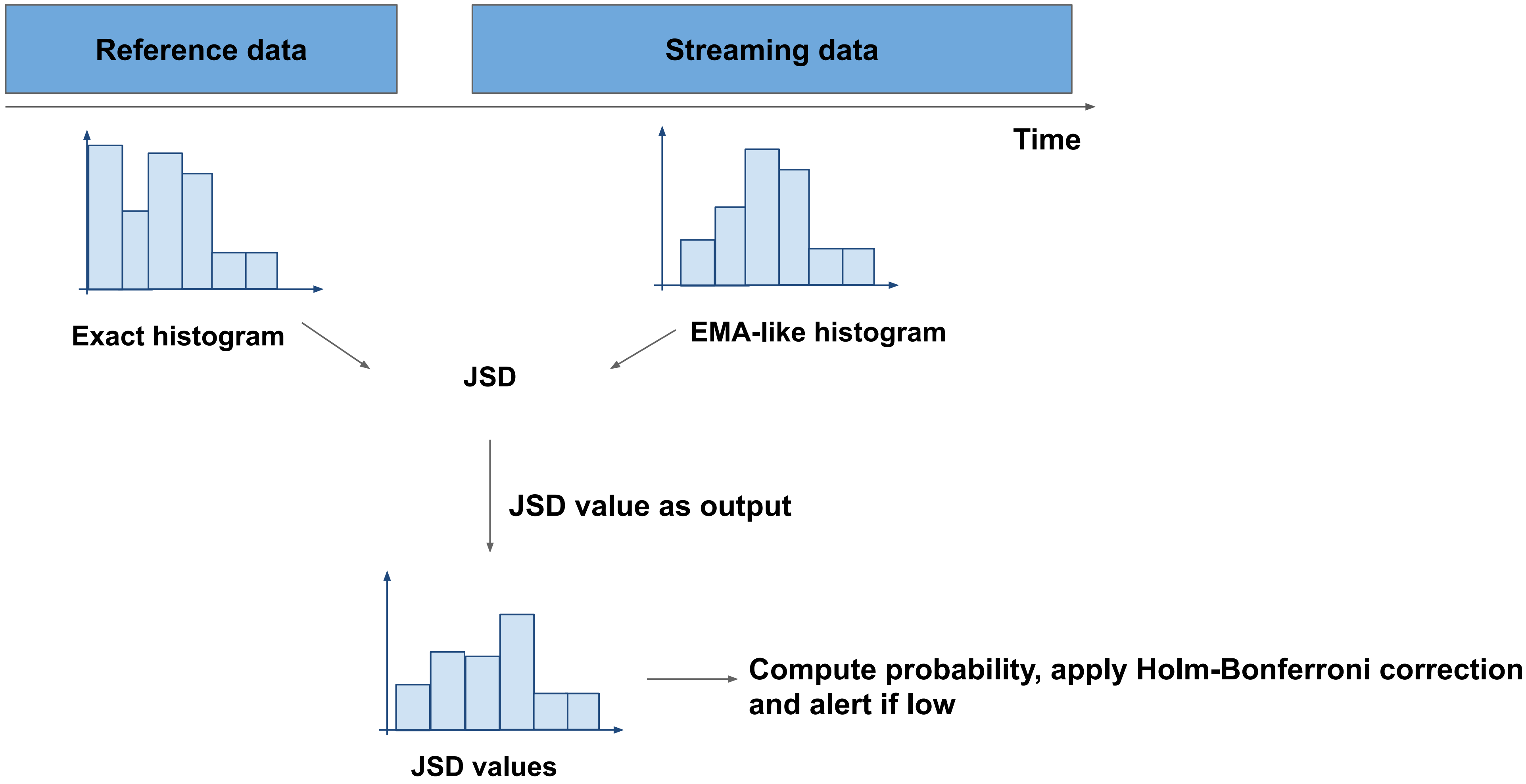
The Holm-Bonferroni method

Multiple testing correction

- To solve this, we apply the **Holm-Bonferroni multiple test correction** [12].

- The Holm–Bonferroni method controls the probability to have one or more false positives by **adjusting the rejection criteria for each of the individual hypotheses.**

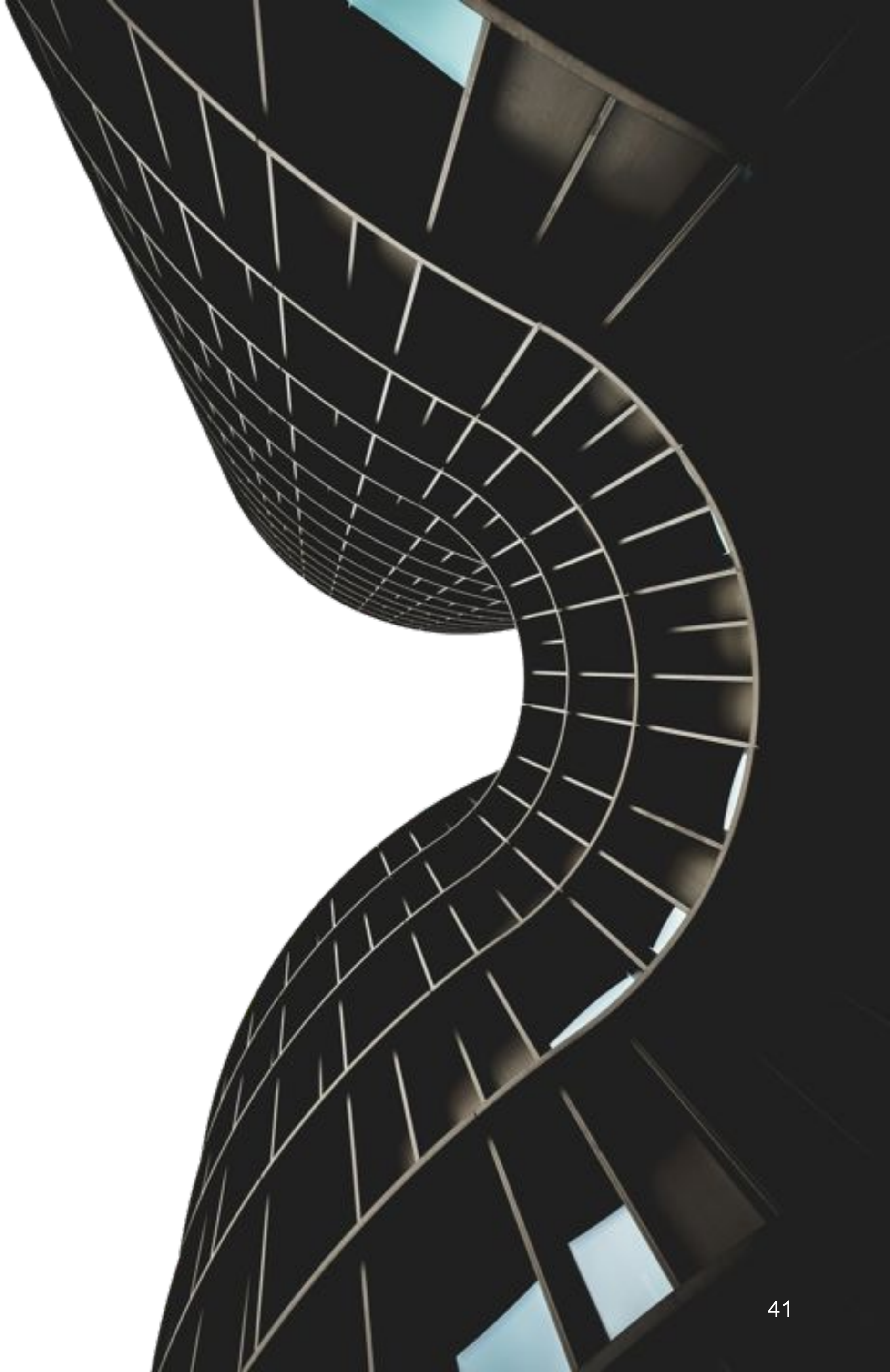
An overview



Reference data
Streaming data
Time
Exact histogram
EMA-like histogram
JSD
JSD value as output
JSD values
Compute probability, apply Holm-Bonferroni correction and alert if low

Experimental Results

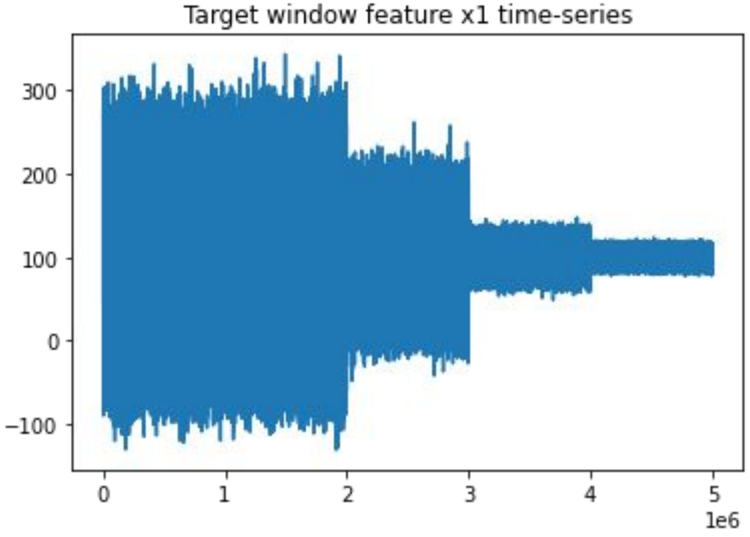
Sanity Checks



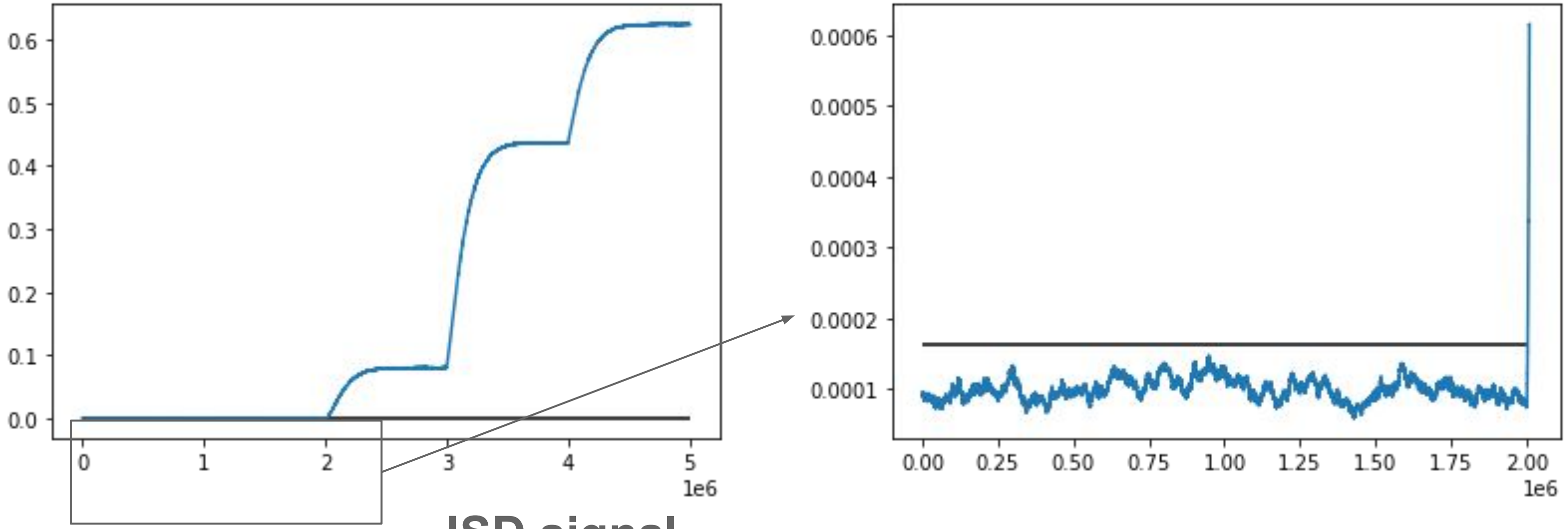
Experimental Results

Sanity check

- For a single feature (x1) scenario on a synthetic dataset.



Target data, simulated stream, feature x1 values
x-axis: event number
y-axis: event value

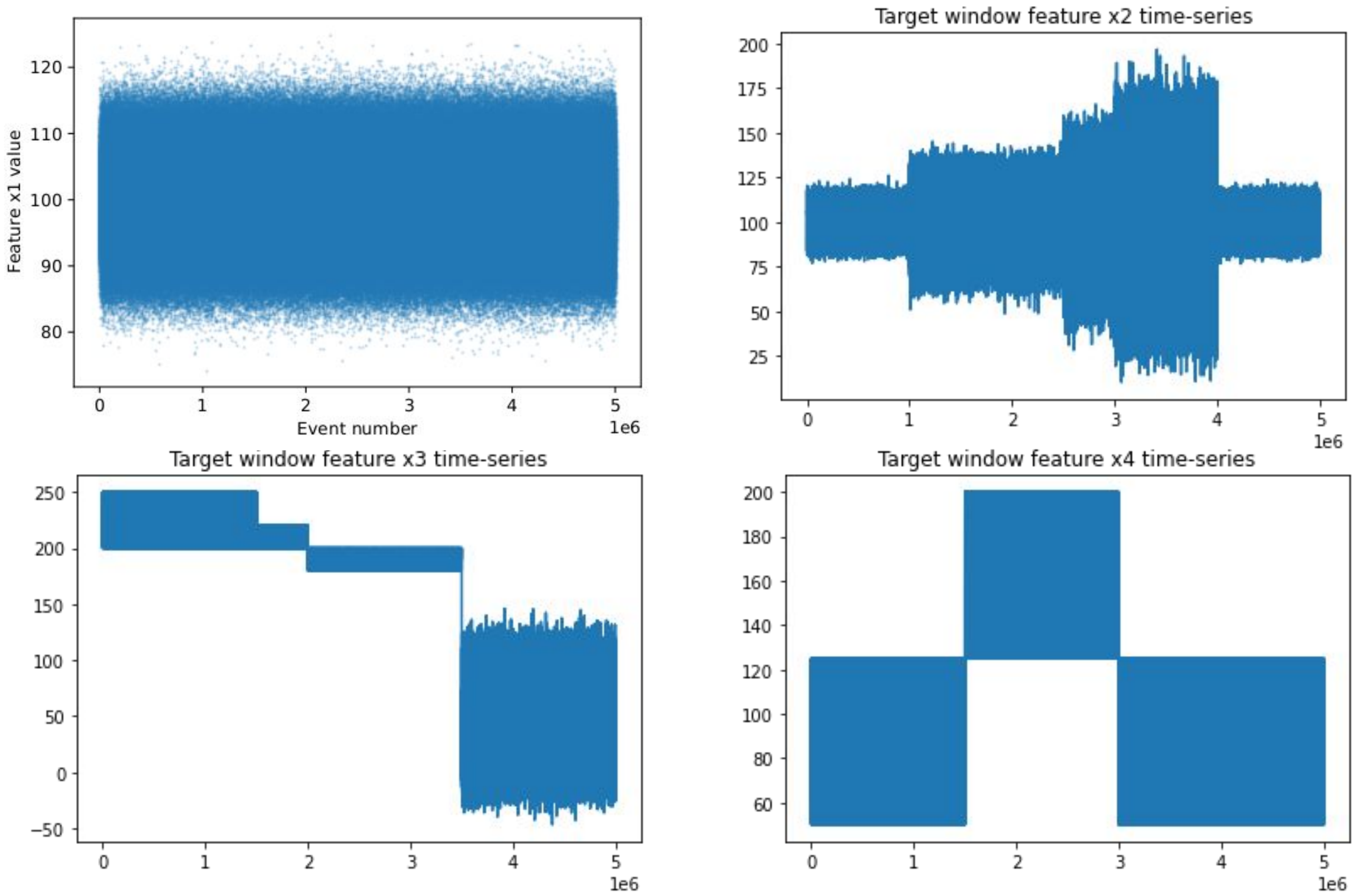


JSD signal
x-axis: event number
y-axis: jsd value between target histogram and reference histogram
horizontal black line: tukey threshold

Experimental Results

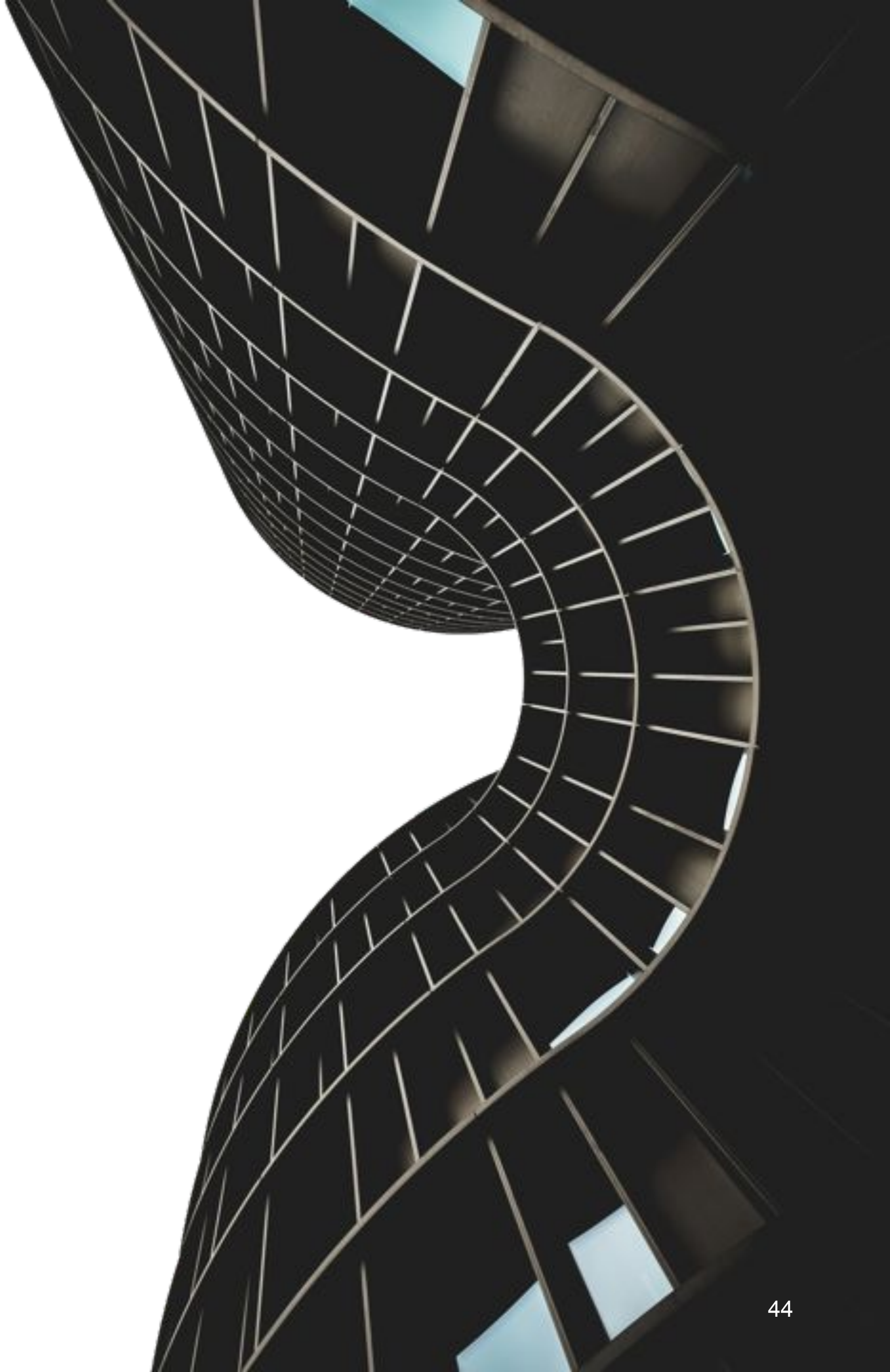
Sanity check

- For a multi-feature scenario (x1, x2, x3, x4).



Experimental Results

Real world data



Experimental Results

Merchant data

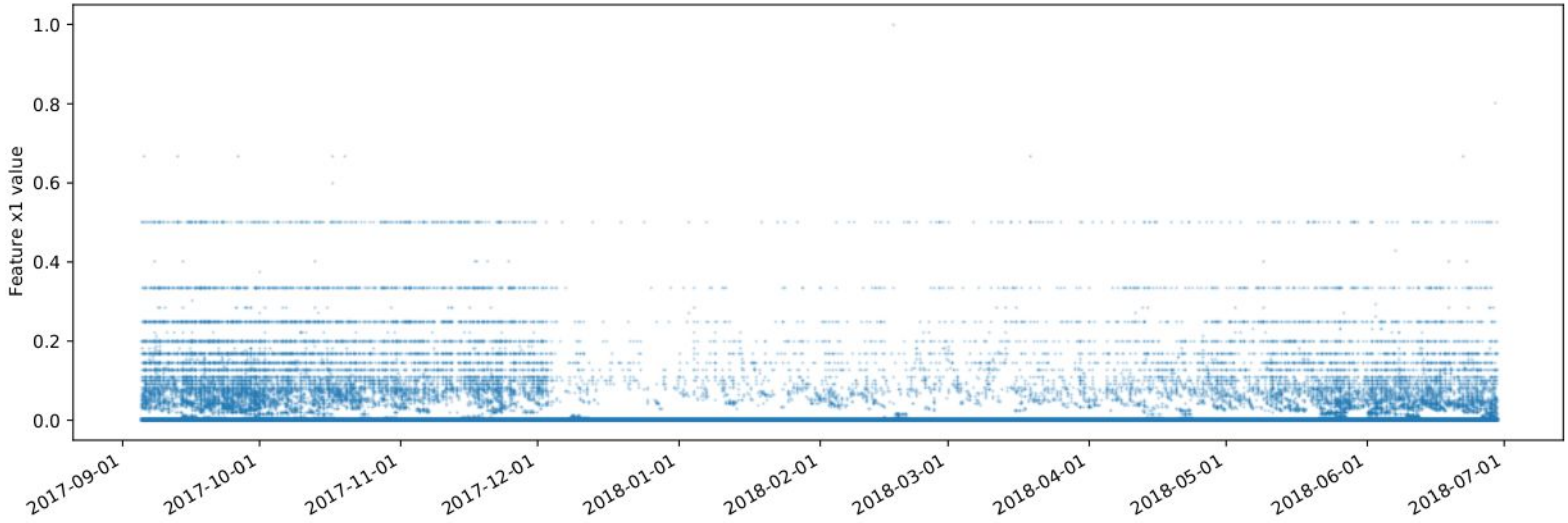
- The table below shows summary statistics for the used reference and target period datasets from a merchant that uses Feedzai's fraud detection system.

Dataset	From	To	Transactions
Reference	Tuesday, September 5, 2017	Friday, June 29, 2018	1,604,509
Target	Friday, June 29, 2018	Sunday, June 30, 2019	4,032,505

Experimental Results

Merchant data

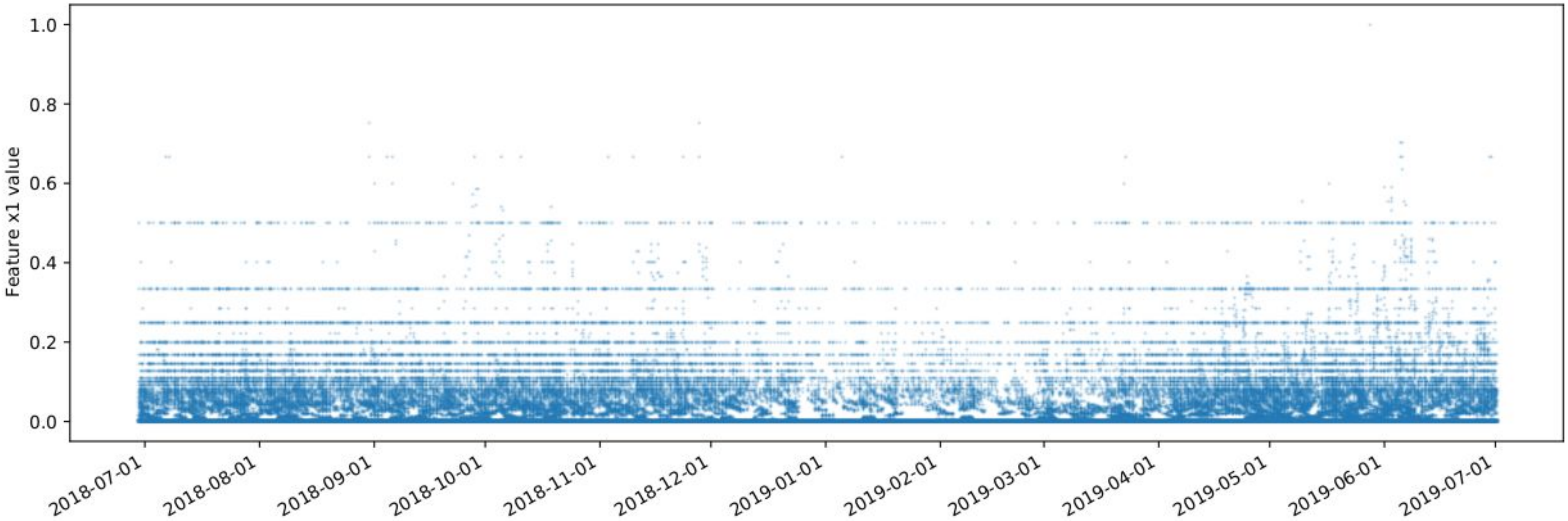
- Feature x1 **reference** time-series.



Experimental Results

Merchant data

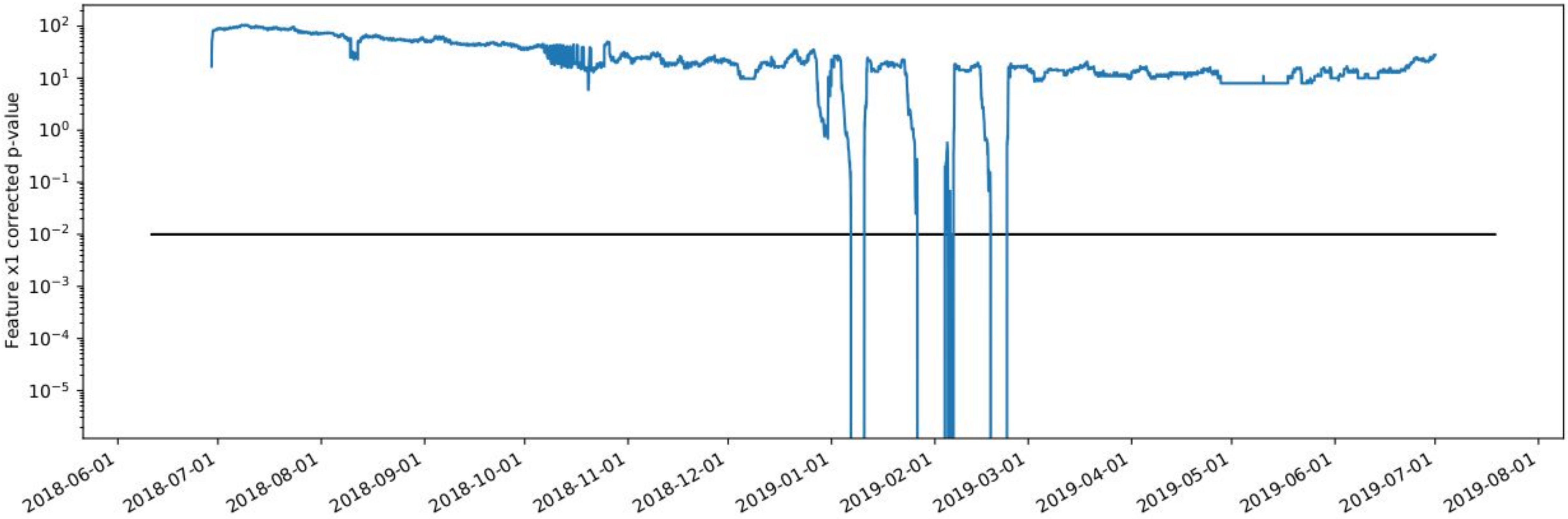
- Feature x1 **target** time-series.



Experimental Results

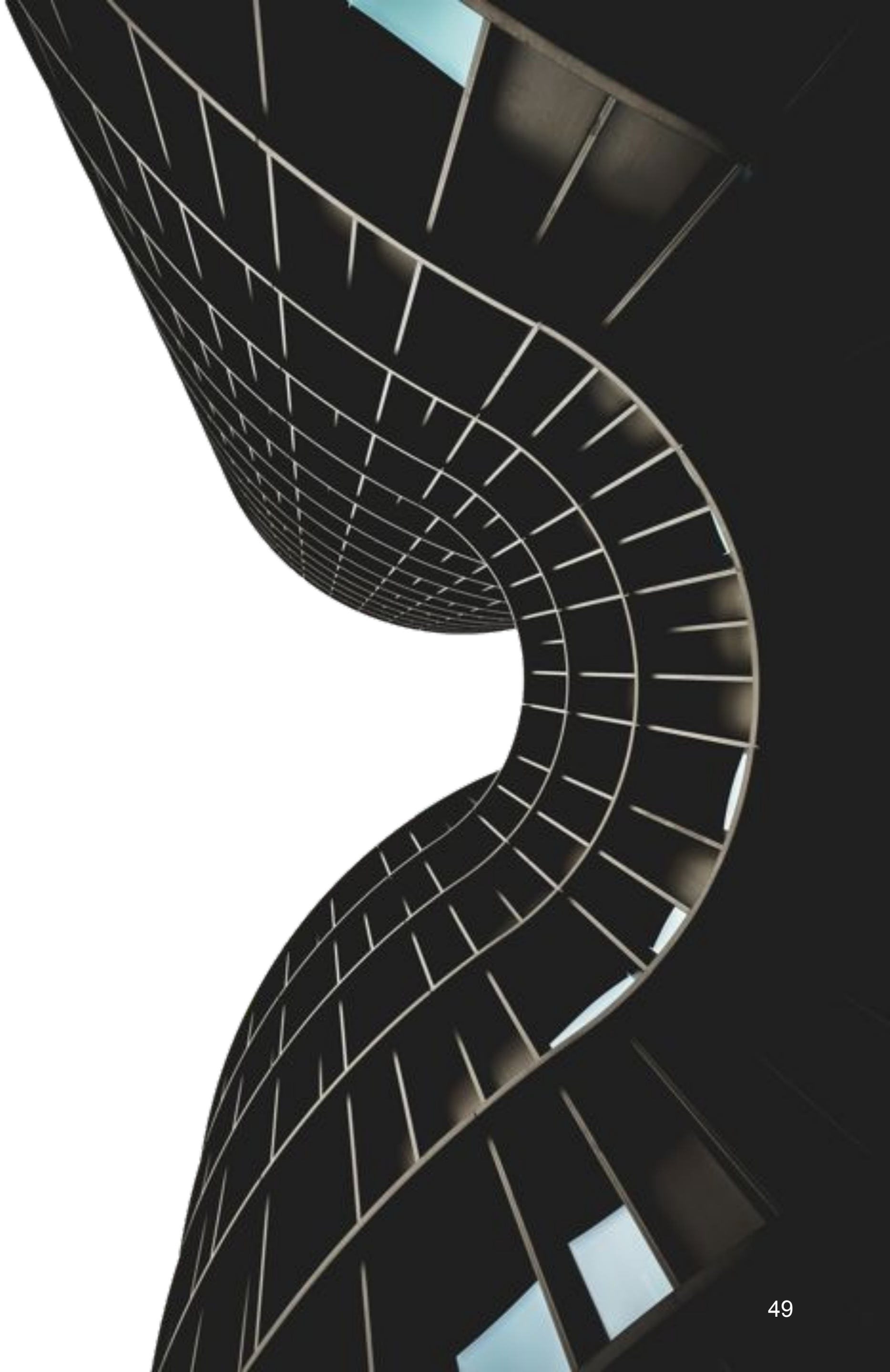
Merchant data

- Feature x1 **corrected p-value** plot and **alarm threshold**.



Experimental Results

Ablation Study



Building an Accurate Divergence Measure Distribution

Ablation study

- In this ablation study we use a **custom built JSD distribution** which is the **best possible fit for the streaming** JSD values.

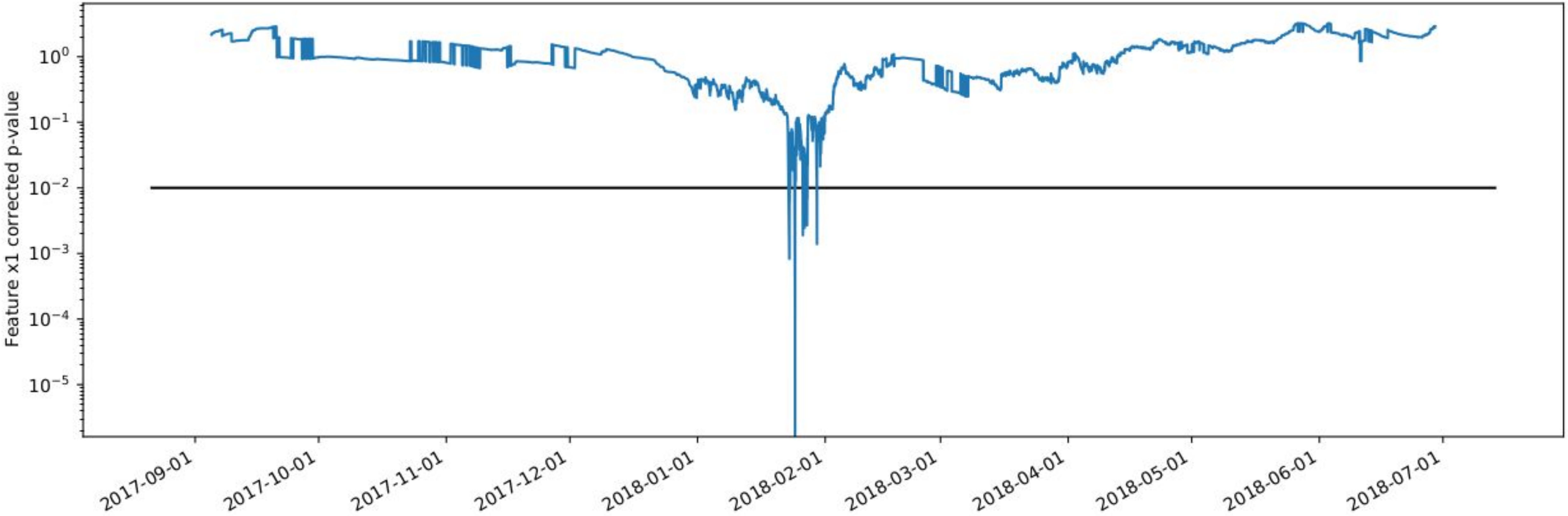
- To do so we **traverse the streaming period, collect all JSDs and build the distribution** from that.

- We then **run our streaming analysis once again using this accurate distribution**.

Building an Accurate Divergence Measure Distribution

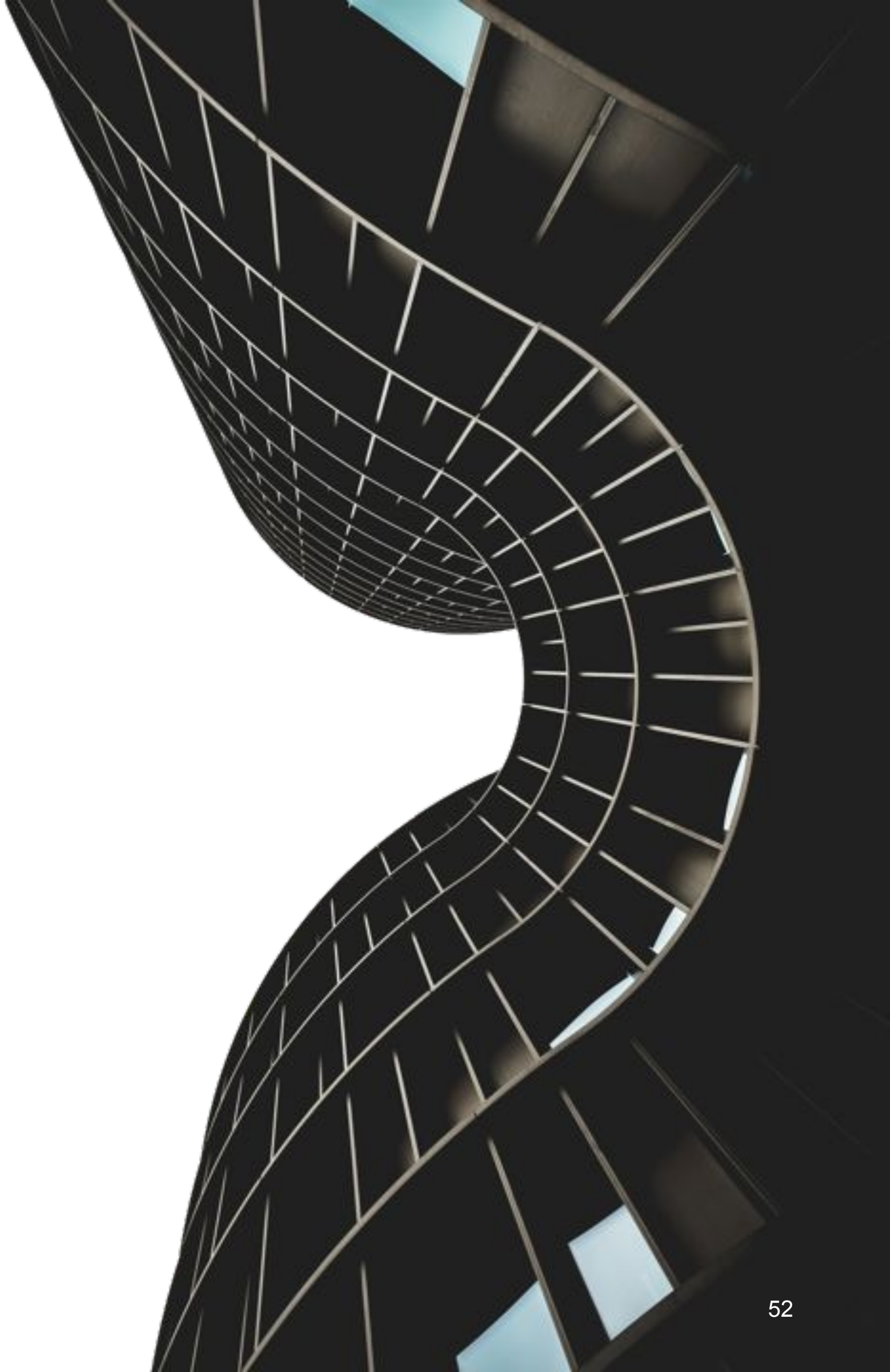
Ablation study

- Feature x1 **corrected p-value** plot and **alarm threshold**.



feedzai

Conclusions



feedzai

Conclusions

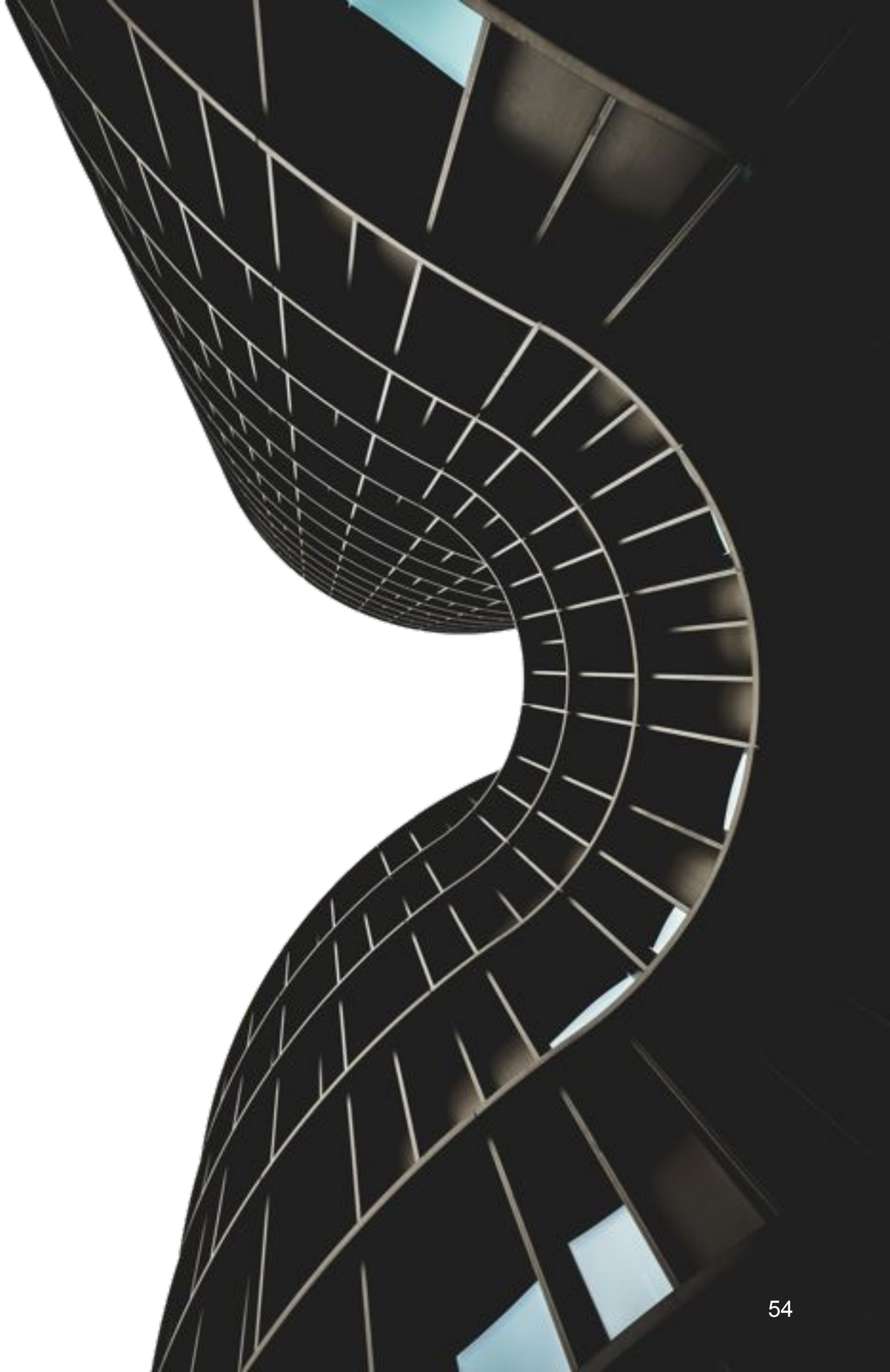
- **Single and multi-feature synthetic data induced anomalies were all reported**. We know this because we had the ground-truth (the one we created).

- On **real data**, we **don't have a ground truth**, but at least **visually the results seem to make some sense**.

- In our **ablation studies**, where we **remove the batch distribution** and **use a perfect fit**, we got **very few and short lasting alerts** (the 99th percentile JSD values) as expected.

feedzai

Contributions



Contributions

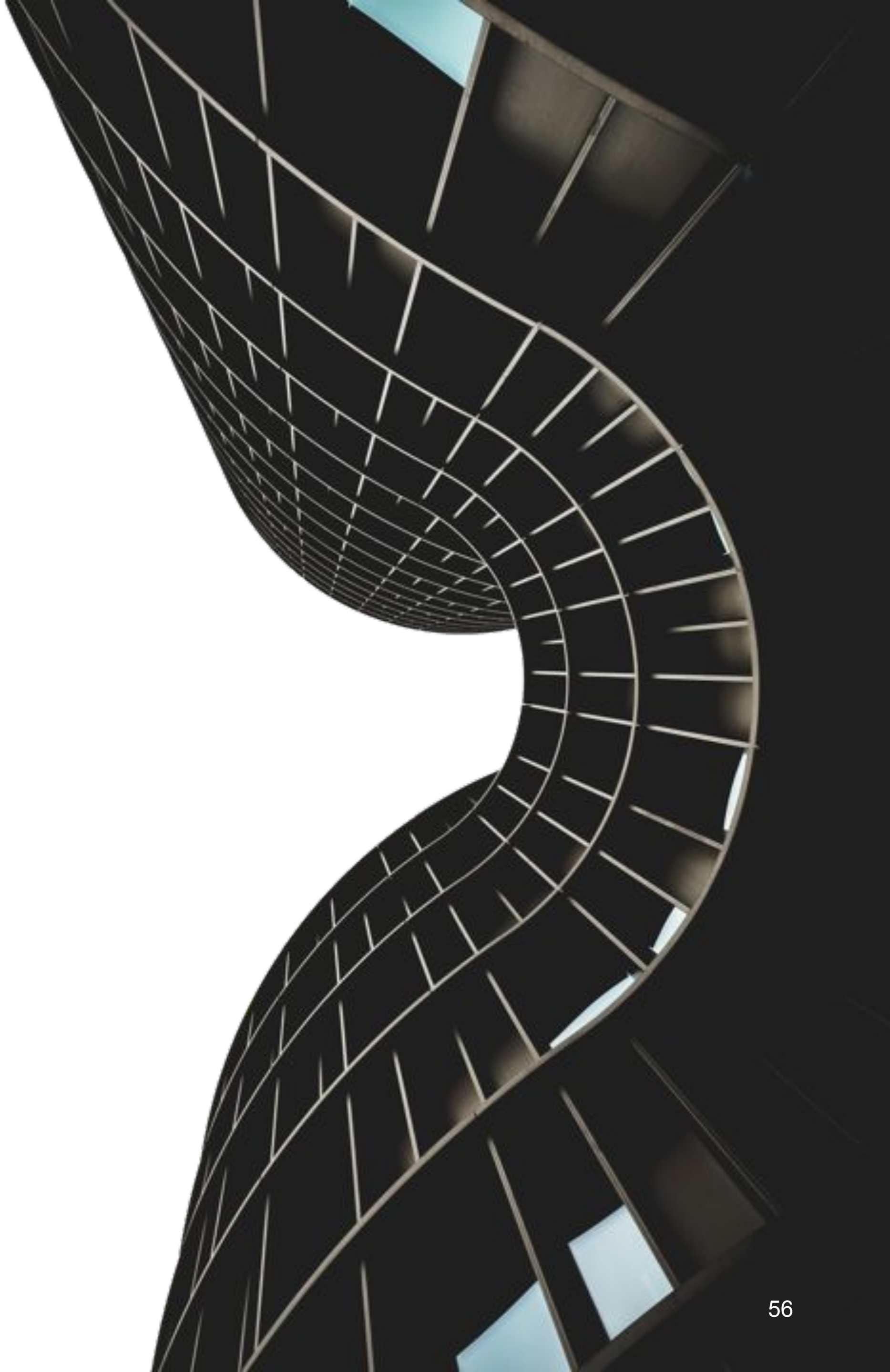
1. A **two-phased** and **two-windowed univariate subsequence outlier detection method** (taxonomy of [13]) that is **lightweight** and works for **real-time stream monitoring**.

2. A **histogram aggregation** based on **EMAs.**

3. A **set of synthetic datasets and experiments** where the method in question accurately detected anomalies.

4. **Experiments on real data** where we provide insights and **possible hypotheses** to test for **future work**.

Future Work



Future Work

1. Experiment making **more samples to build a more accurate JSD distribution**.

2. Test with **other divergence measures**. We suggest trying the Wasserstein [14] distance.

3. Test with **other multiple test correction methods**.

Thank you

feedzai

References

[1] A. Tsymbal, “The problem of concept drift: Definitions and related work,” 05 2004.

[2] J. a. Gama, I. Žliobaitundefined, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” ACM Comput. Surv., vol. 46, Mar. 2014.

[3] F. Pinto, M. O. P. Sampaio, and P. Bizarro, “Automatic model monitoring for data streams,” 2019

[4] J. Lin, “Divergence measures based on the shannon entropy,” IEEE Transactions on Information Theory, vol. 37, no. 1, pp. 145–151, 1991.

[5] J. E. Everett, “The exponentially weighted moving average applied to the control and monitoring of varying sample sizes,”WIT Transactions on Modelling and Simulation, vol. 51,pp. 3–13, 2011.

[6] J. S. Hunter, “The Exponentially Weighted Moving Average,”Journal of Quality Technology, vol. 18, no. 4, pp. 203–210, 1986.

[7] M. B. Perry, The Exponentially Weighted Moving Average. American Cancer Society, 2011.

[8] M. Aickin and H. Gensler, “Adjusting for multiple testing when reporting research results: the bonferroni vs holm methods.,” American journal of public health, vol. 86 5, pp. 726–8, 1996.

[9] T. Dickhaus, “Simultaneous statistical inference,” in Springer Berlin Heidelberg, 2014.

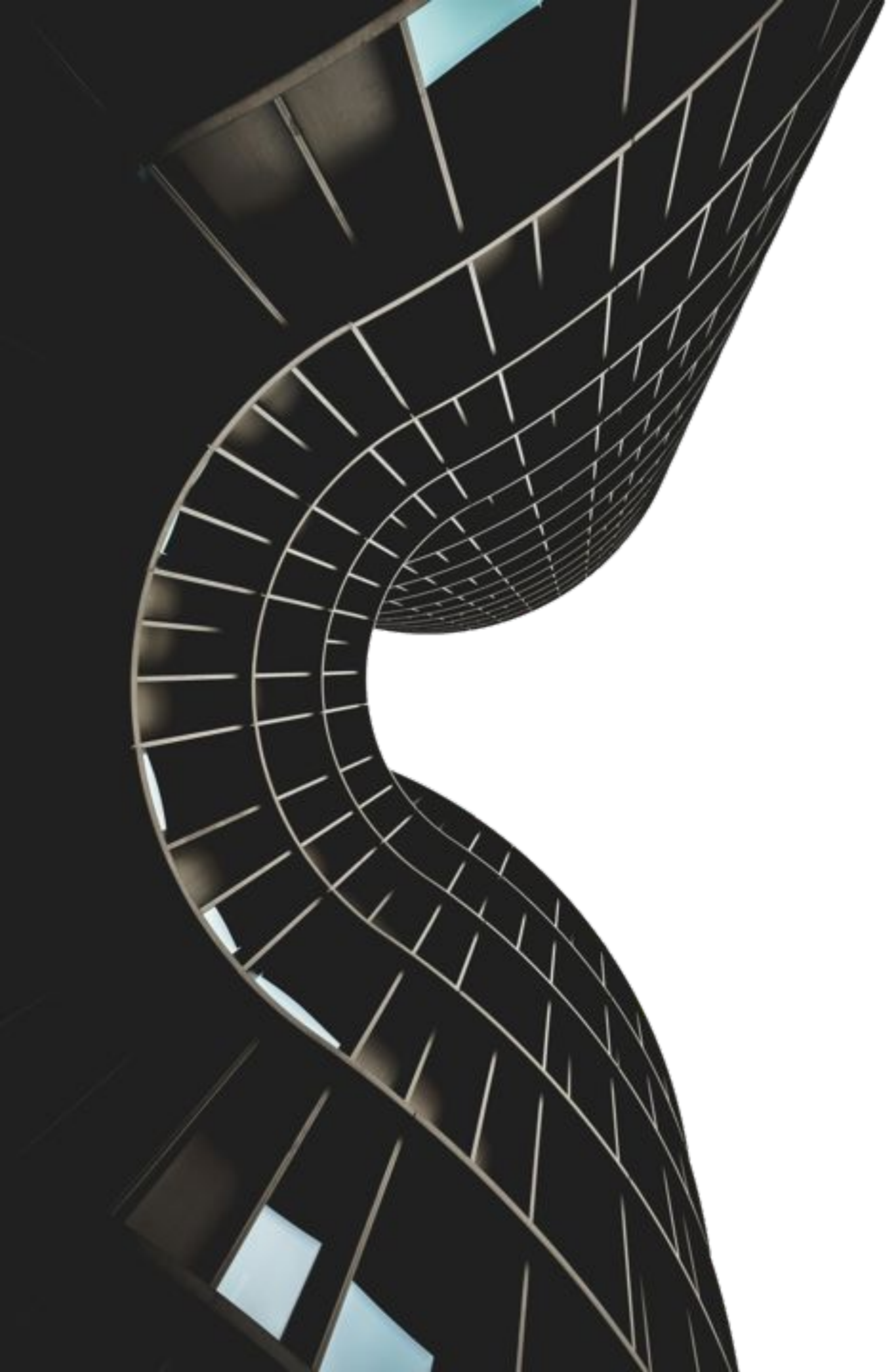
[10] R. G. Miller, “Simultaneous statistical inference,” 1966.

[11] G. I. Webb and F. Petitjean, “A multiple test correction for streams and cascades of statistical hypothesis tests,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16, (New York, NY, USA), p. 1255–1264, Association for Computing Machinery, 2016.

[12] S. Holm, “A simple sequentially rejective multiple test procedure,” 1979.

[13] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, “A review on outlier/anomaly detection in time series data,” ArXiv, 2020.

[14] M. Hazewinkel, The Encyclopaedia of Mathematics, vol. 5, pp. 102–140. 01 2000.



feedzai

Lightweight Real-Time Feature Monitoring

Master Thesis Defense

July 24th, 2020

Auxiliary slides for committee questions

Batch analysis

A two-phased method

- **Why compute multiple sample histograms?**

  To **encode the distributions of smaller time periods** since there may be pattern changes **within the large reference** period.

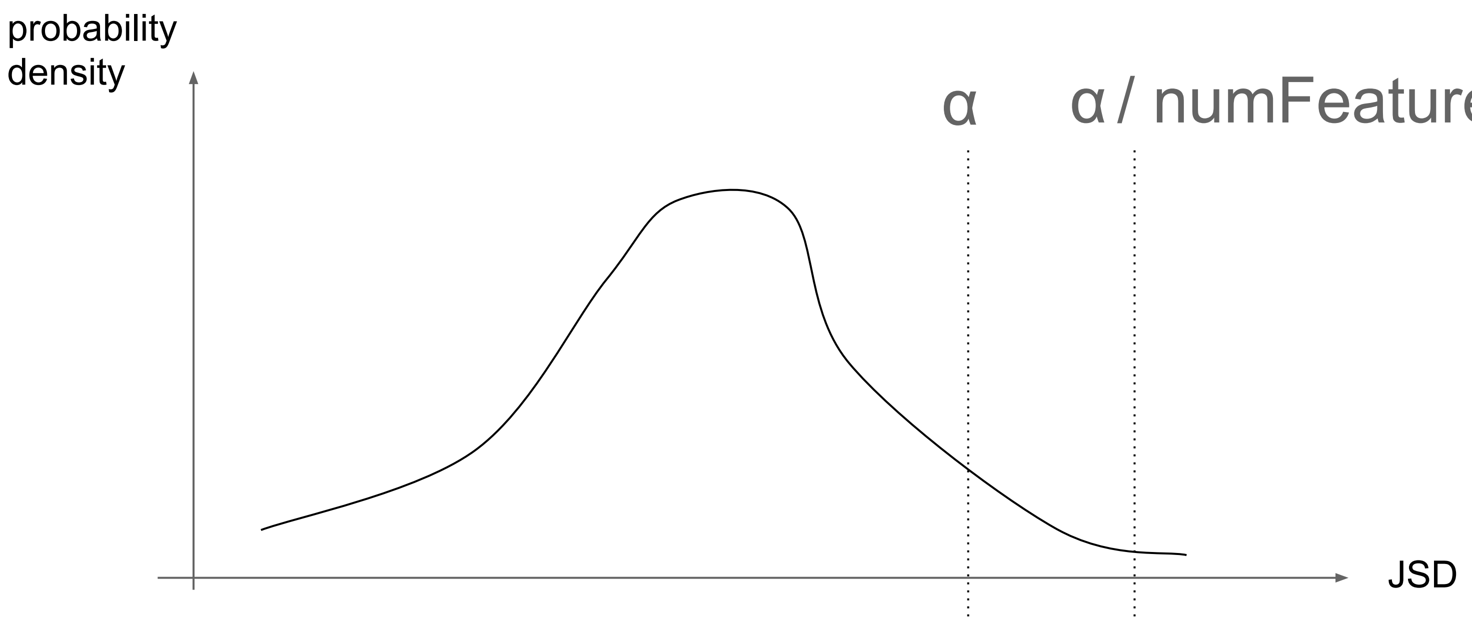
- **Why an approximated histogram?**

  To **mimic the streaming EMA histogram** we will incrementally maintain in the streaming environment.

Final remarks about the method

Minimum number of samples

- We need to **estimate reliably the upper tail of the JSD distribution**. With the Holm-Bonferroni correction the upper tail of the distribution is the **region α/numFeatures**



Final remarks about the method

Minimum number of samples

- γ is the **probability of having zero** sample JSD **points in the upper tail**.

- According to the [3], the **minimum number of samples** to make is:

$$n_{samples} = \frac{\log \gamma}{\log \left(1 - \frac{\alpha}{n_{features}}\right)}$$

Final remarks about the method

Minimum number of samples

$$n_{samples} = \frac{\log \gamma}{\log\left(1 - \frac{\alpha}{n_{features}}\right)}$$

- For **180 features, γ = 1% and α = 1%**, we have around **~83k samples**

Final remarks about the method

Sample size

- We set a **half-life value** and then **multiply it by a constant factor of 4** to get the size of the samples to make.

- Processing **4xhalf-life events** means the **oldest event's contribution** is approximately 6%.

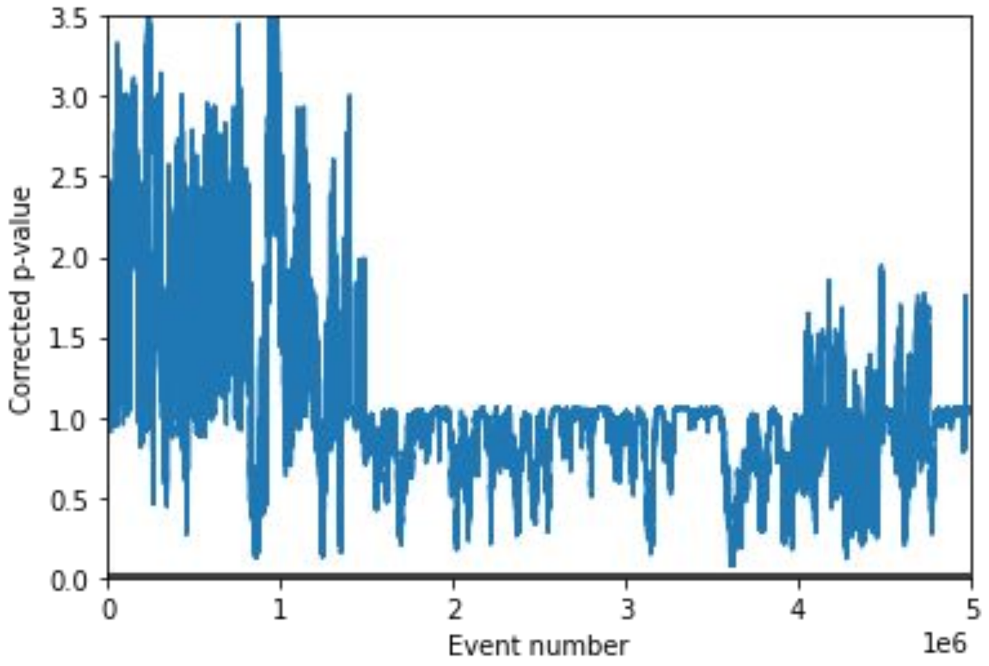
$$\left(2^{-\frac{1}{n_{1/2}}}\right)^{4n_{1/2}} = 2^{-\frac{4n_{1/2}}{n_{1/2}}} = 2^{-4} \approx 0.06 = 6\%$$

- We consider **6% low enough** and don't process any more events.

On synthetic data

Experimental Results

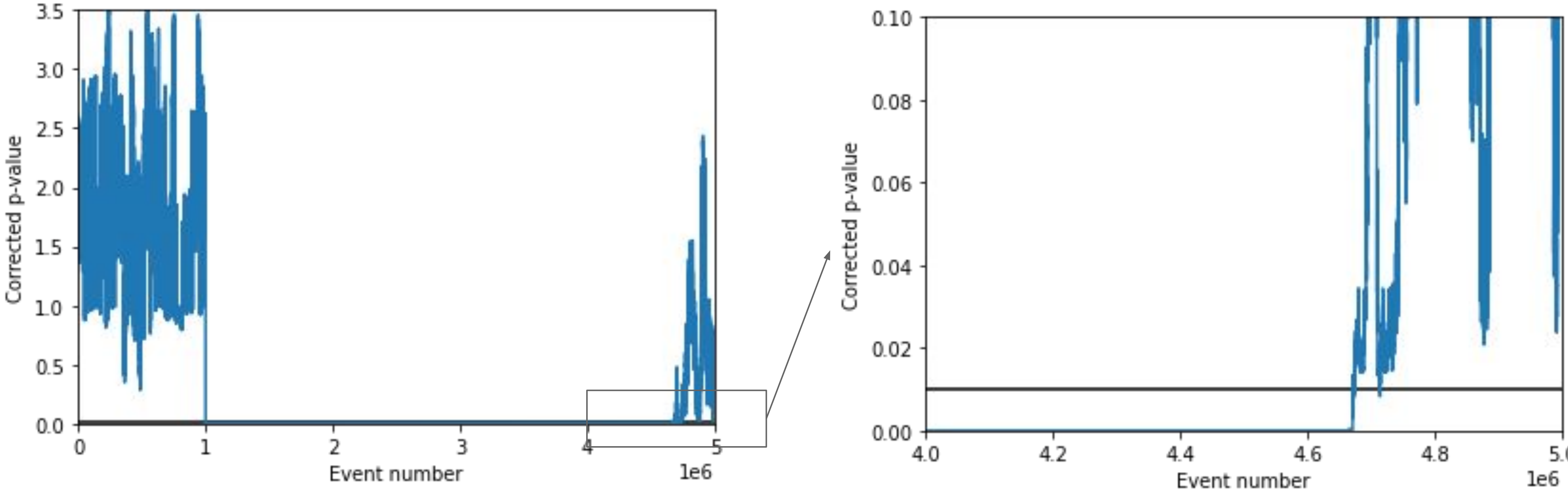
- x1 never diverged from its reference, so we expect to see the corrected Holm-Bonferroni p-values above the 1% probability threshold



On synthetic data

Experimental Results

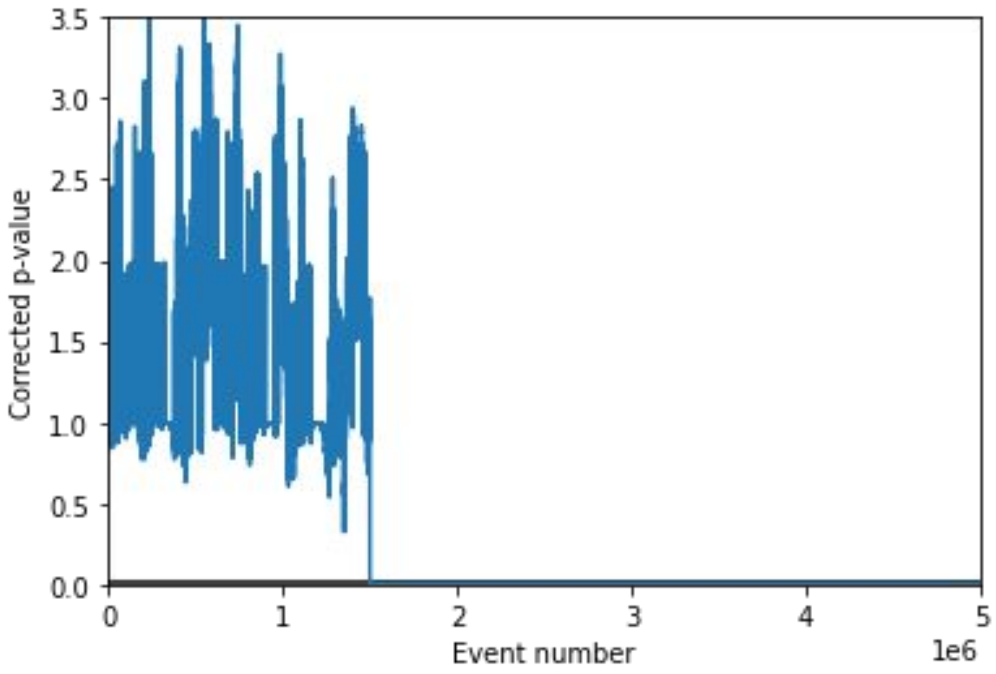
- x2 diverges quite a lot but eventually returns to its original distribution: as a result we see it drops below the threshold until it resumes the reference distribution



On synthetic data

Experimental Results

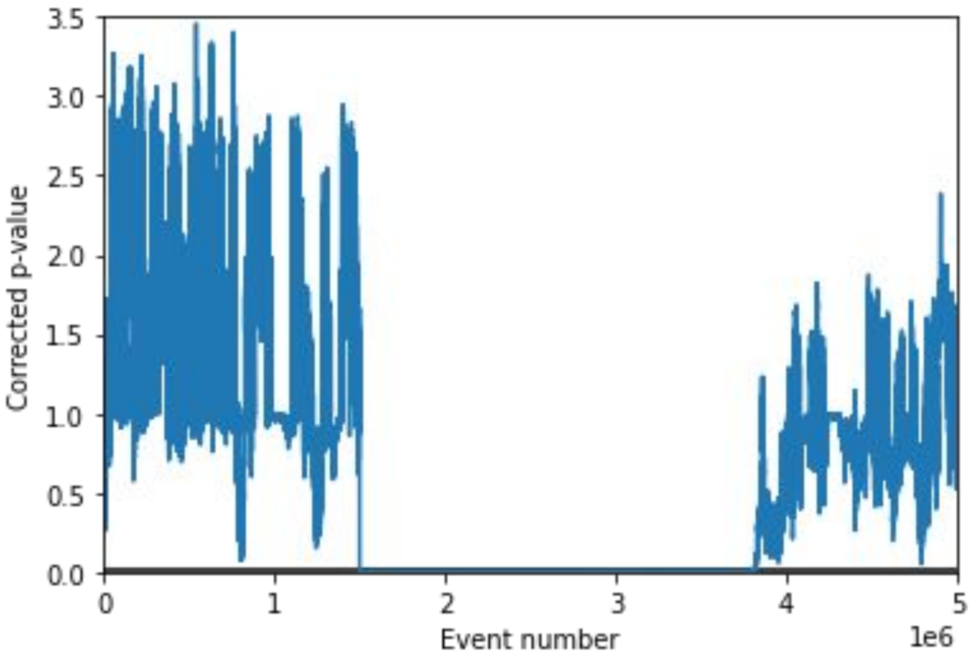
- x3 diverges quite a lot as well and never resumes its reference distribution



On synthetic data

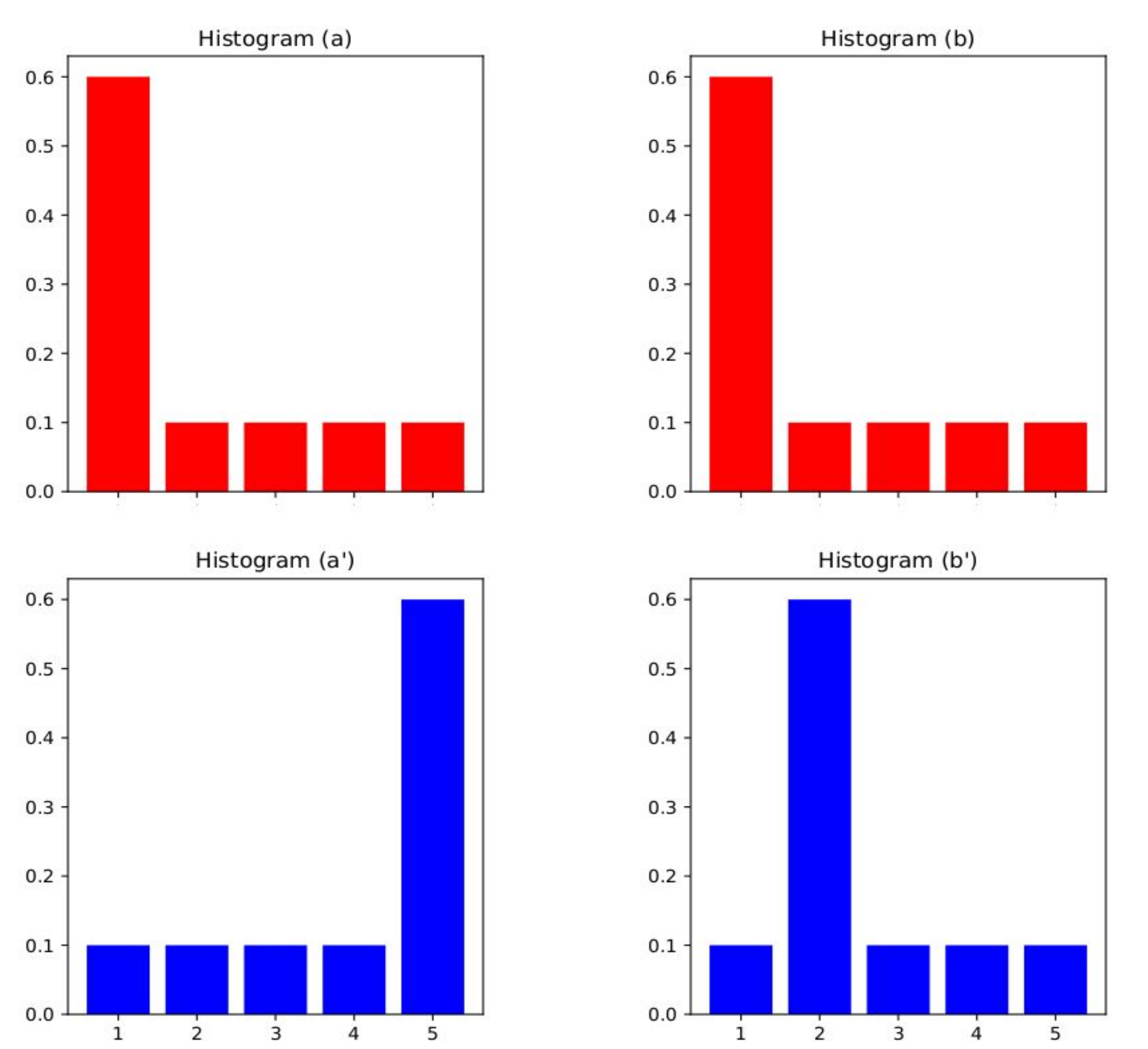
Experimental Results

- x4 diverges but returns to its original distribution



Future Work

3. Test with other divergence measures then JSD.
   We suggest trying the **Wasserstein distance** next.

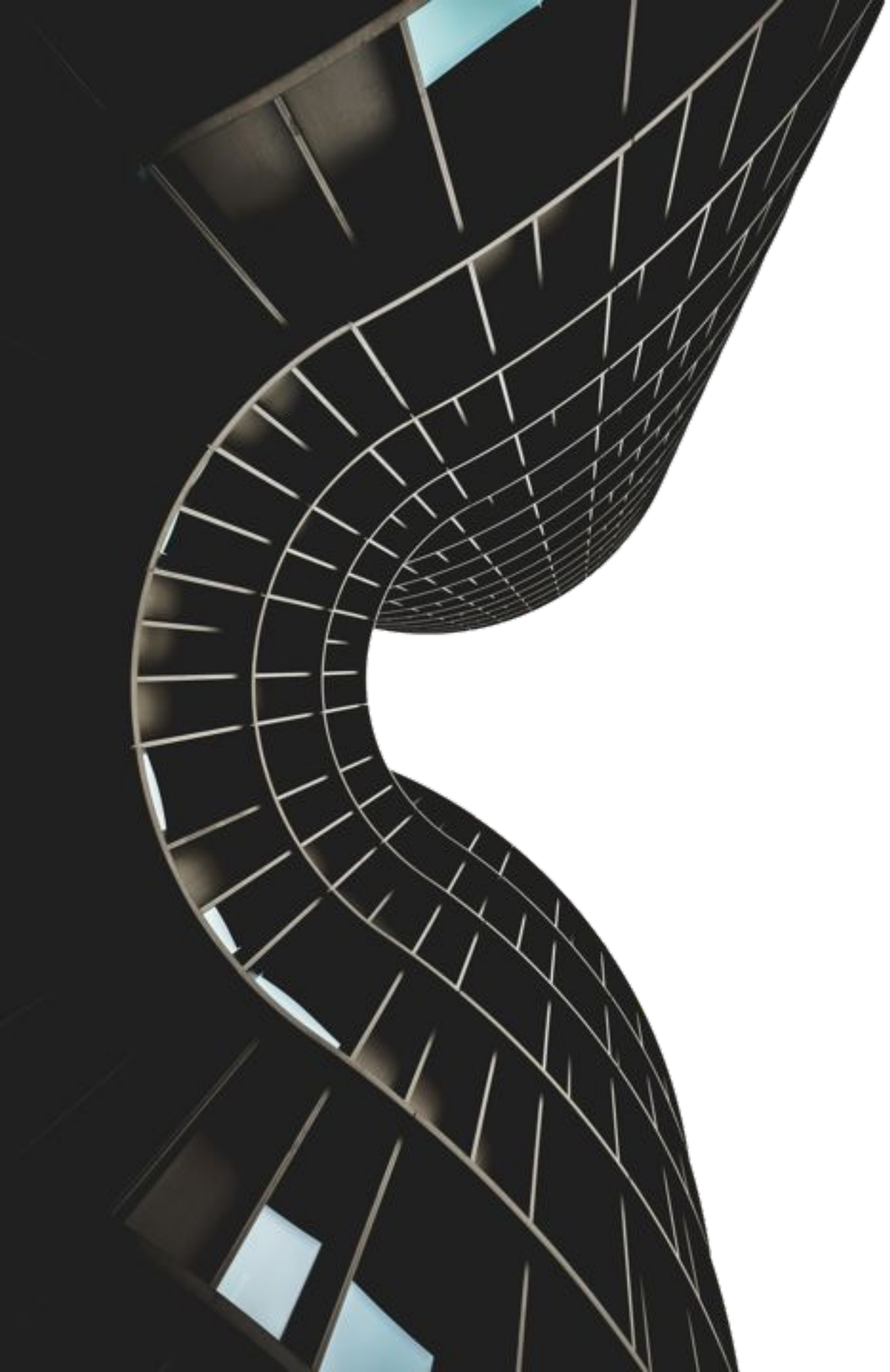


Future Work

- **JSD value is the same** for both distribution pairs while **Wasserstein is not**.

- **Wasserstein** takes into account the **amount of probability mass that has to be transported.**

Distribution #1	Distribution #2	Wasserstein Output	Jensen–Shannon Output
a	a’	2.0	0.4451
b	b’	0.5	0.4451



feedzai

Lightweight Real-Time Feature Monitoring

Master Thesis Defense

July 24th, 2020