MacroBase: Prioritizing Attention in Fast Data

Peter Bailis, Edward Gan,
Samuel Madden, Deepak
Narayanan, Kexin Rong,
Sahaama Suri

abundant data, scarce attention

abundant data, scarce attention

data is increasingly too big for manual inspection

abundant data, scarce attention

data is increasingly too big for manual inspection

Twitter, LinkedIn, Facebook, Google: log 12M+ events/s

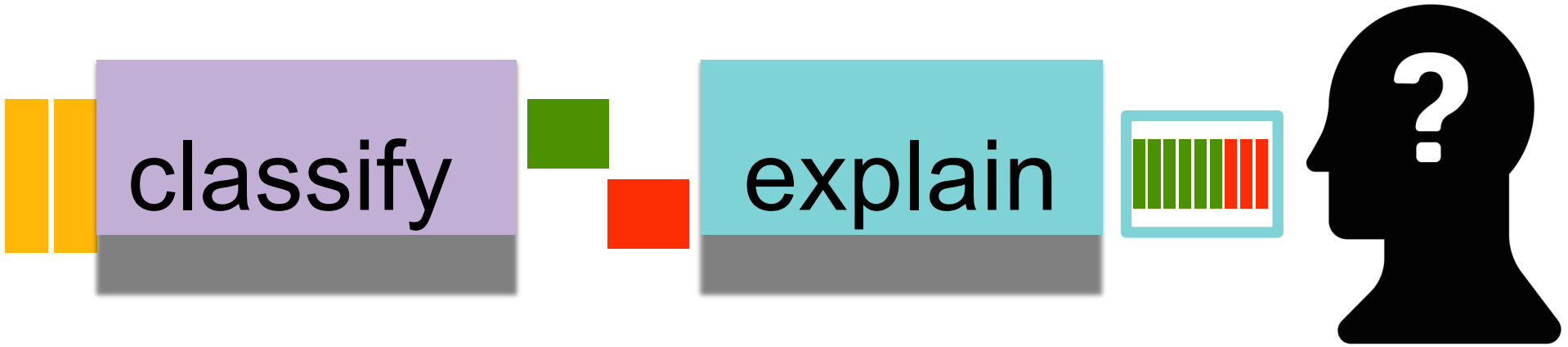
projected 40% year-over-year growth (e.g., via IoT)

abundant data, scarce attention

data is increasingly too big for manual inspection

- Twitter, LinkedIn, Facebook, Google: log 12M+ events/s
- projected 40% year-over-year growth (e.g., via IoT)
- today's operators say:
- < 6% of this data is ever accessed after ingest

the key to fast data
combine: classify and explain



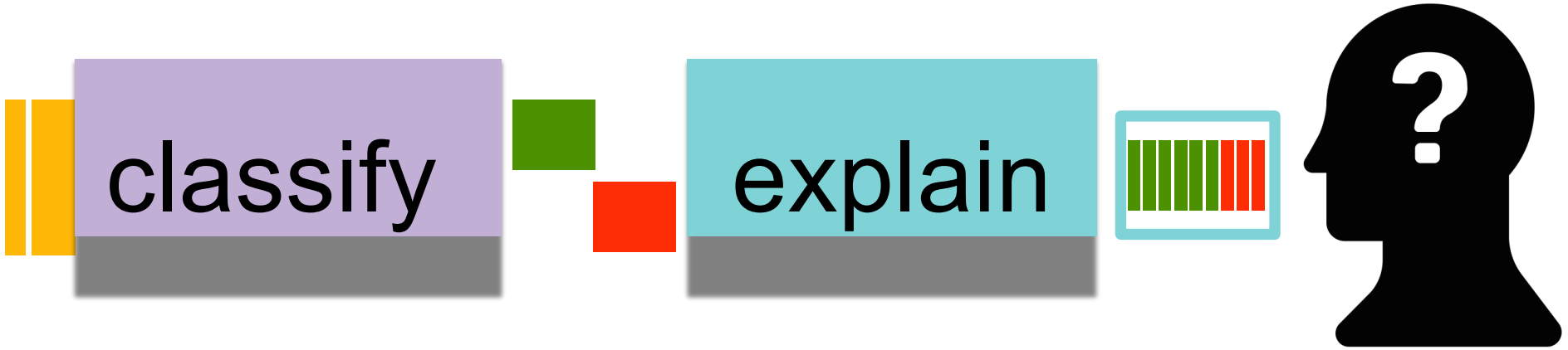
how should we do it?

dataflow (alone) is not enough

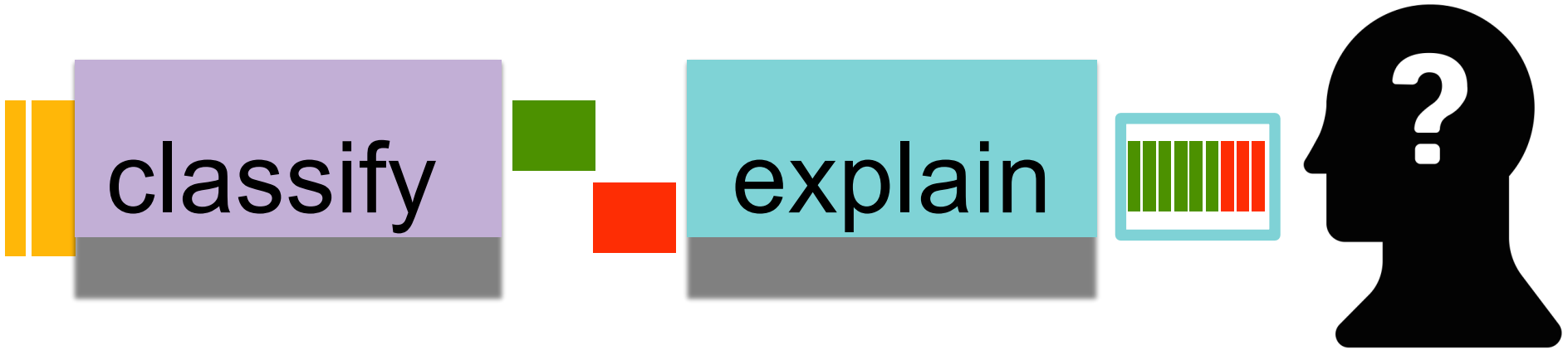
dataflow: a substrate, not a complete solution

missing:
scalable, modular operators
for prioritizing attention via
classification and explanation

macrobase: a fast data system



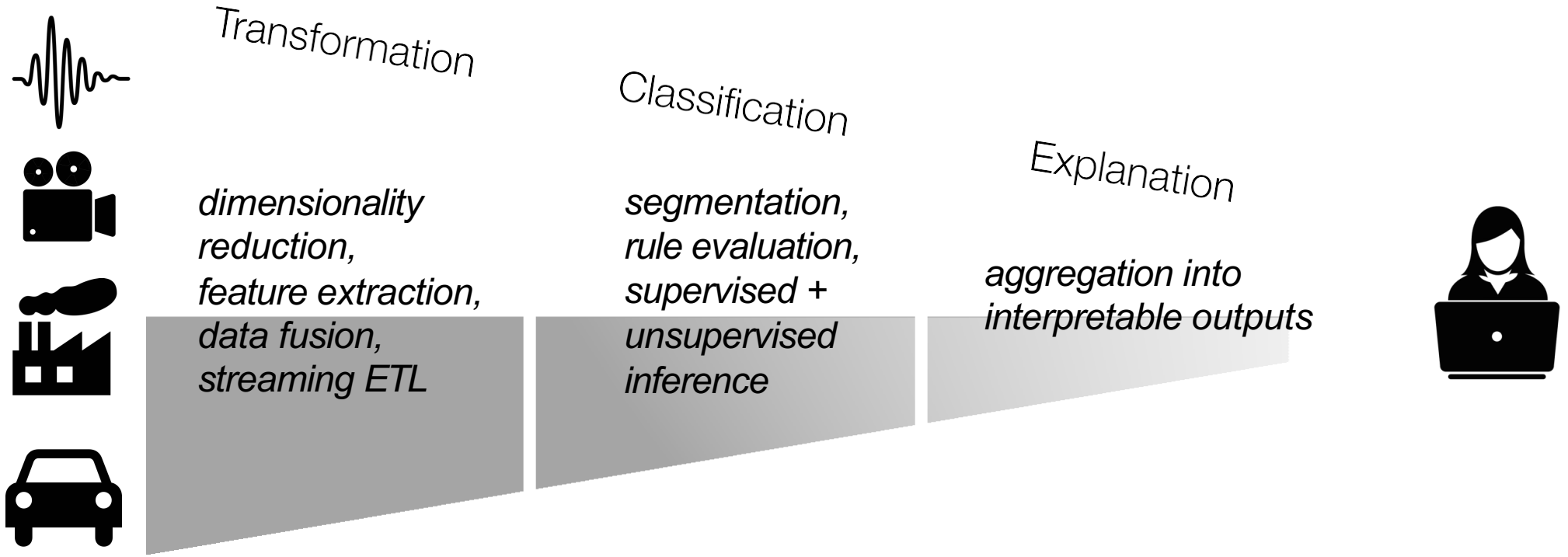
a system providing
fast, reusable, modular operators
for classification and explanation
prioritizing attention in fast data



key: make this combo fast

MacroBase System: Prioritize Attention

Execute *cascades* of operators that transform, filter, aggregate the stream



MacroBase: Data points and operators

Operators

1) **Ingestion:** Stream<Point>
2) **Transformation:** Stream<Point> → Stream<Point>
3) **Classifier:** Stream<Point> → Stream<label, Point>
4) **Explanation:** Stream<label, Point> → Stream<Explanation>

Data Types

1) **Point:** (array<double> metrics, array<varchar> attributes)
2) **Explanation:** (array<varchar> attributes, stats statistics)

Overview

Ingestion
Requires necessary stream ordering done before ingestion.

Transformation
Domain specific. User performs this without modifying later stages.

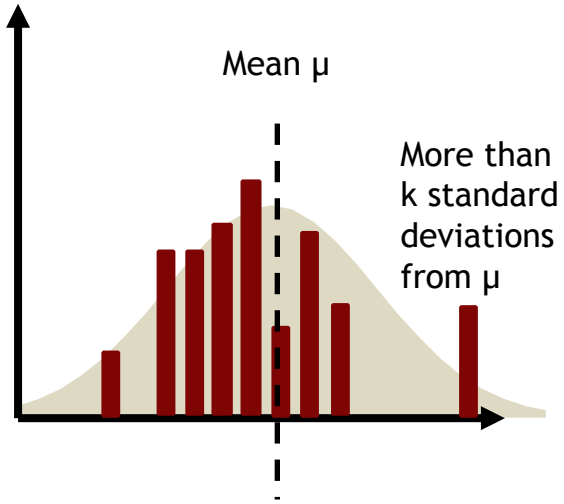
Classification
Label each point according to its input metrics.

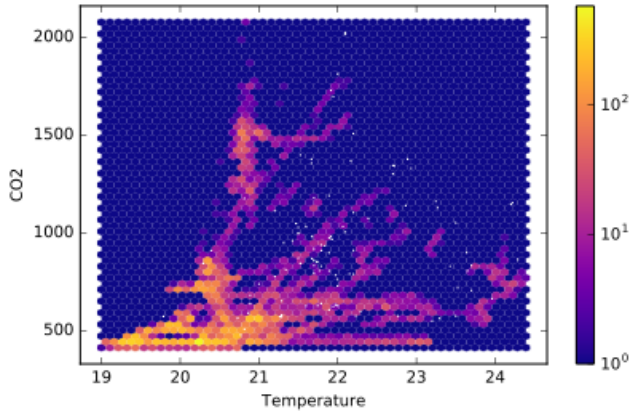
Explanation
Return attribute value combination that is common among outlier points
But not common among inliers.

End-to-end optimizations

1. **Adaptable damped reservoir**
   Novel sampling in streaming data context.
2. **Cardinality Imbalance**
   Optimized way to generate explanations explanations
3. **Amortized Maintenance Counter**
   Similar attributes in fast data streams.

Classification





Default: identify unlikely data points (e.g., via density estimation)

unsupervised density based classifiers

Default Classification

Distance of a point from center of the distribution:

Center = Median
Unit distance = Mean Absolute Deviation (MAD)

Score = $\frac{point - Center}{unit\ distance} = \frac{point - median}{MAD}$

For multivariate: Minimum Covariance Determinant (MCD)

Identifies most extreme points based on percentile based cutoff

Classification: Streaming Execution

Idea: Maintain Sample of data

Traditional approach:
Reservoir Sampling: Choosing k samples from a very large stream

Adaption of Reservoir Sampling: **Adaptive Damped Reservoir (ADR)**

- Model retraining is done using ADR on input stream.
- Outliers are maintained in ADR, which is used to periodically compute the score quantile.

Algorithm 1 ADR: Adaptable Damped Reservoir

given: k: reservoir size $\in \mathbb{N}$; r: decay rate $\in (0, 1)$
initialization: reservoir $R \leftarrow \{\}$; current weight $c_w \leftarrow 0$
function OBSERVE(x: point, w: weight)
    $c_w \leftarrow c_w + w$
    if $|R| < k$ **then**
        $R \leftarrow R \cup \{x\}$
    else with probability $\frac{k}{c_w}$
        remove random element from R and add x to R
function DECAY( )
    $c_w \leftarrow r \cdot c_w$

Explanation

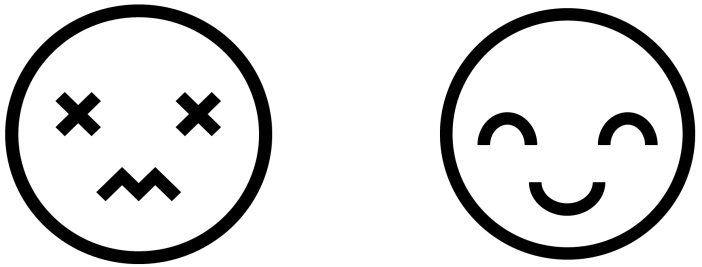
Errors	Non-Errors
{iPhone7, Canada}	{iPhone8, USA}
{iPhone7, USA}	{iPhone7, USA}
{iPhone8, Canada}	{iPhoneX, USA}
{iPhone7, USA}	{iPhone7, USA}
{iPhone8, Canada}	{iPhone7, USA}
	{iPhone8, USA}
	{iPhone7, USA}
	{iPhone7, USA}

Canada may have a problem!

Explain classification results that describes attributes common to outliers but relatively uncommon to inliers.

Explanation

Relative Risk



error non-error

$$\frac{P(\text{error} \mid \text{canada})}{P(\text{error} \mid \text{not canada})}$$

Explain classification results that describes attributes common to outliers but relatively uncommon to inliers.

Default: relative risk calculation based on data attributes

Risk Ratio for various attributes

A.	B
B.	C
A	D
A	A
A	B
	D
	C
	B
	B
	E
	B

Naive Solution ?

Optimization: Individual ratio vs Set

Optimization: Exploit Cardinality imbalance

1. Search for attributes in outliers (= O) with minimum risk ratio.
2. Compute attribute combinations over O (= C).
3. Filter C by risk in inliers is less than r.

Note: Inliers>> outliers

Streaming Explanation

Maintain count of frequent items = Heavy hitters

Heavy hitters → Approximate risk ratio and occurrence of attributes.

Traditional Solution: Space saving algorithm with exponential decayed settings

Problem: Performance. (Update time is more. We have to update frequency for every point.)

Streaming Explanation

Optimized Heavy hitters: Much larger space

Algorithm 3 AMC: Amortized Maintenance Counter

given: $\varepsilon \in (0,1)$; r: decay rate $\in (0,1)$
initialization: C (item $\rightarrow$ count) $\leftarrow \{\}$; weight $w_i \leftarrow 0$
function OBSERVE(i: item, c: count)
    $C[i] \leftarrow w_i + c$ if $i \notin C$ else $C[i] + c$
function MAINTAIN( )
    remove all but the $\frac{1}{\varepsilon}$ largest entries from C
    $w_i \leftarrow$ the largest value just removed, or, if none removed, 0
function DECAY( )
    decay the value of all entries of C by r
    call MAINTAIN( )

Evaluation

Results: MacroBase on 4 Intel Xeon CPU with 12 cores each and 1 TB of RAM.

Default minimum occurrence/support: 0.1%
Default minimum risk: 3
Target outlier percentile: 1%
ADR and AMC sizes of 10K
Decay range of 0.01 for every 100K points.

Synthetic dataset accuracy

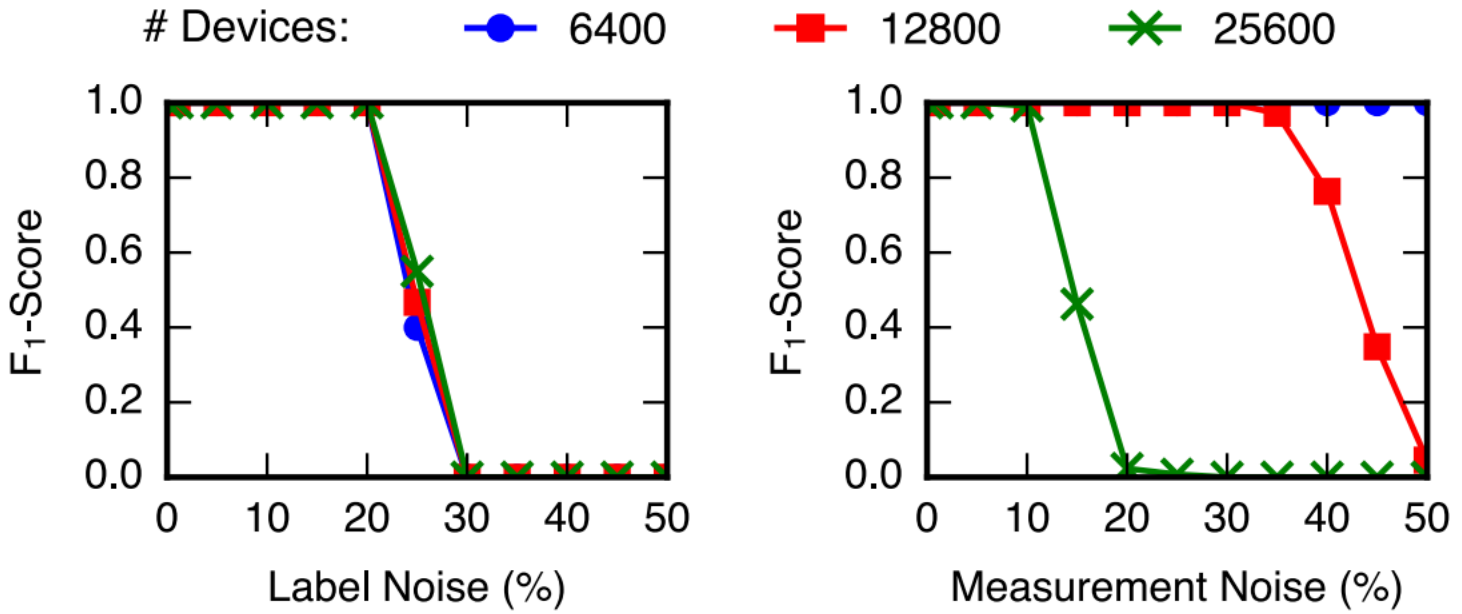
Data from # devices.

- Inlier distribution N (10,10)
- Outlier distribution N (10,10)
- Total Points: 1M

$$\text{F1 Score} = 2.\frac{Precision.recall}{precision+recall}$$

label noise: Randomly assigning readings from the outlier distribution to inlying devices and vice-versa.

Measurement noise: Randomly assigning a proportion of both outlying and inlying points to a third, uniform distribution over the interval [0,80].



Adaptivity

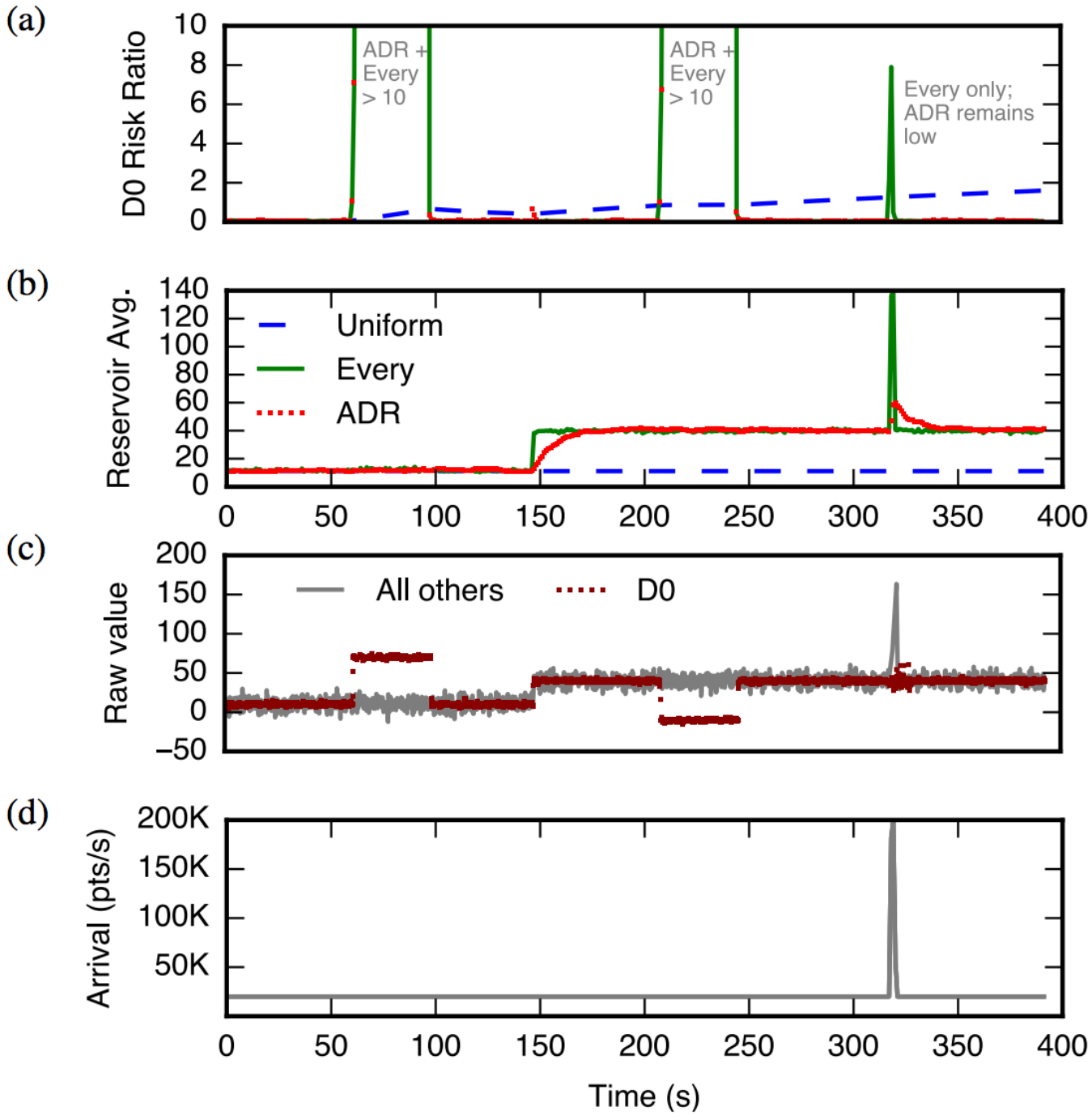
Time evolving data stream repr

Change underlying data distribution
and **Variable arriving rate**

100 Devices

1. **0-50 sec, All devices: Gaussian N(10,10)**
2. **50-100 sec, D0 : Gaussian N(70,10)**
3. **100-150 sec, D0 return to N(10,10)**
4. **150-300 sec, all devices N(40,10)**
5. **-----**

Uniform: Uniform Sampling Reservoir
Every: Per-tuple exponentially decaying
ADR: Adaptive Damped Reservoir



Other Comparisons & Results

Effect of cardinality aware explanation: 3 X improvement

Effect of AMC (Heavy hitters): 500X improvements due to extra space (space is 1000X more than space saving algorithm)

End to end runtime breakdown of time

- % of time in training MAD, scoring points and generating explanation

Discussed Case Studies

1. Supervised classifier
2. Transformation

Early Production Usage

CAMBRIDGE MOBILE TELEMATICS

"MacroBase discovered a rare issue with the CMT application and a device-specific battery problem. Consultation and investigation with the CMT team confirmed these issues as previously unknown..."

[Bailis et al., SIGMOD 2017]

Conclusion