CS 8803-MDS

Human-in-the-loop Data Analytics

Lecture 16

10/19/22

Logistics

First progress report due this Friday (10/21)

- Assignment available on gradescope

Project update on 10/31

Today's class

MacroBase: Prioritizing Attention in Fast Data

Authors: Haotian, Yiheng

Reviewer: Eric

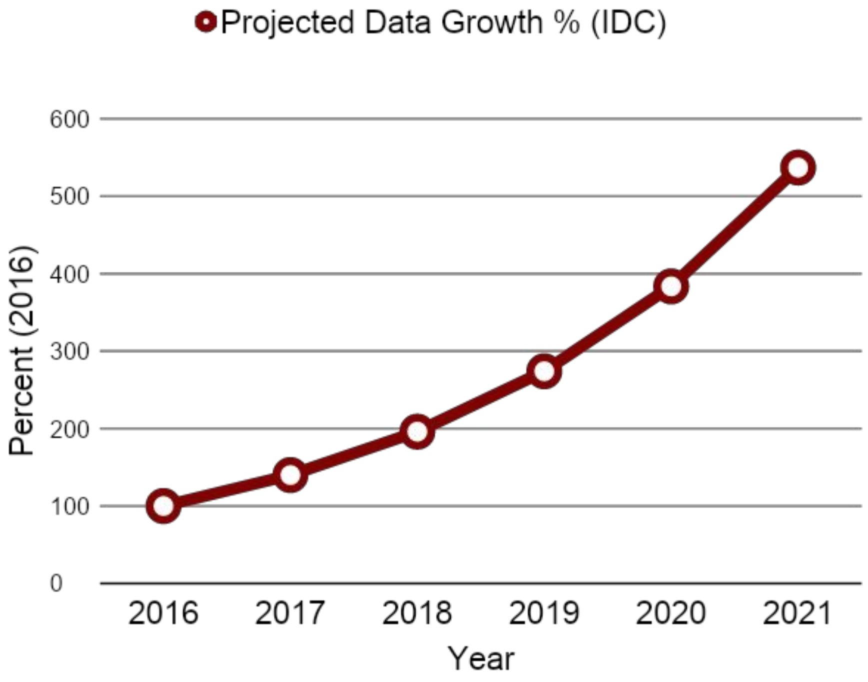
Researcher: Cuong

MacroBase: Prioritizing Attention in Fast Data

SIGMOD/PODS 2017
Chicago, Illinois, USA, May 14-19

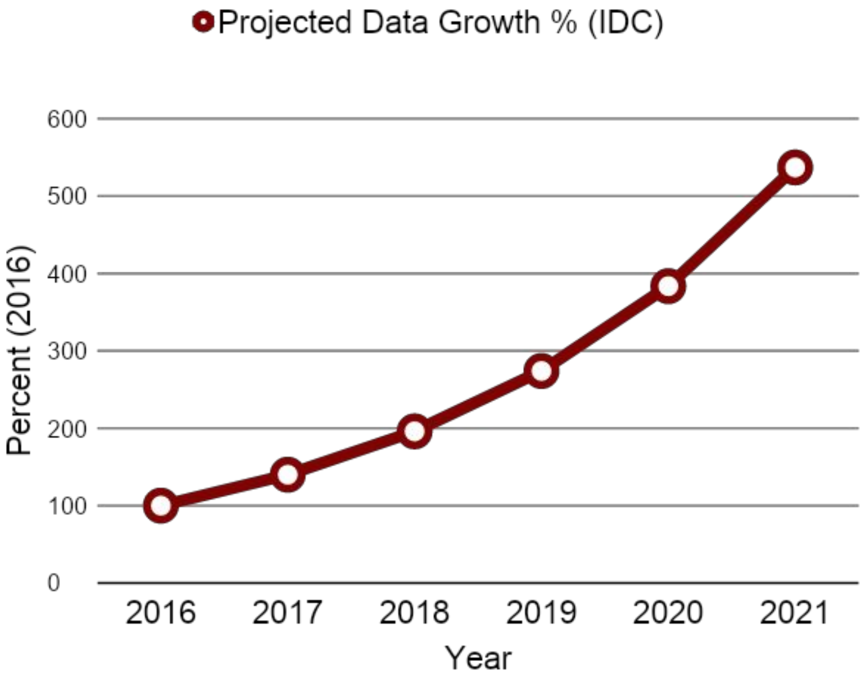
Presenters: Yiheng Mao, Haotian Sun

What’s Happening in Data Analytics



Exponential growth in data volume from 2016 to 2021

What's Happening in Data Analytics



- Data volumes are quickly outpacing human abilities to process them.
- Today, Twitter, LinkedIn, and Facebook each record over 12M events per second.
- Machine-generated data sources are projected to increase data volumes by 40% each year.

Scalable Dataflow Processing Engines in Industry

APACHE HBASE

APACHE STORM™

kafka

APACHE Spark™

ALLUXIO

Spark Streaming

cassandra

Grafana

beam

graphite

Prometheus

APACHE nifi

InfluxDB

Scalable Dataflow Processing Engines in Industry

APACHE HBASE

APACHE STORM™

kafka

APACHE Spark™

ALLUXIO

cassandra

Spark Streaming

Grafana

beam

graphite

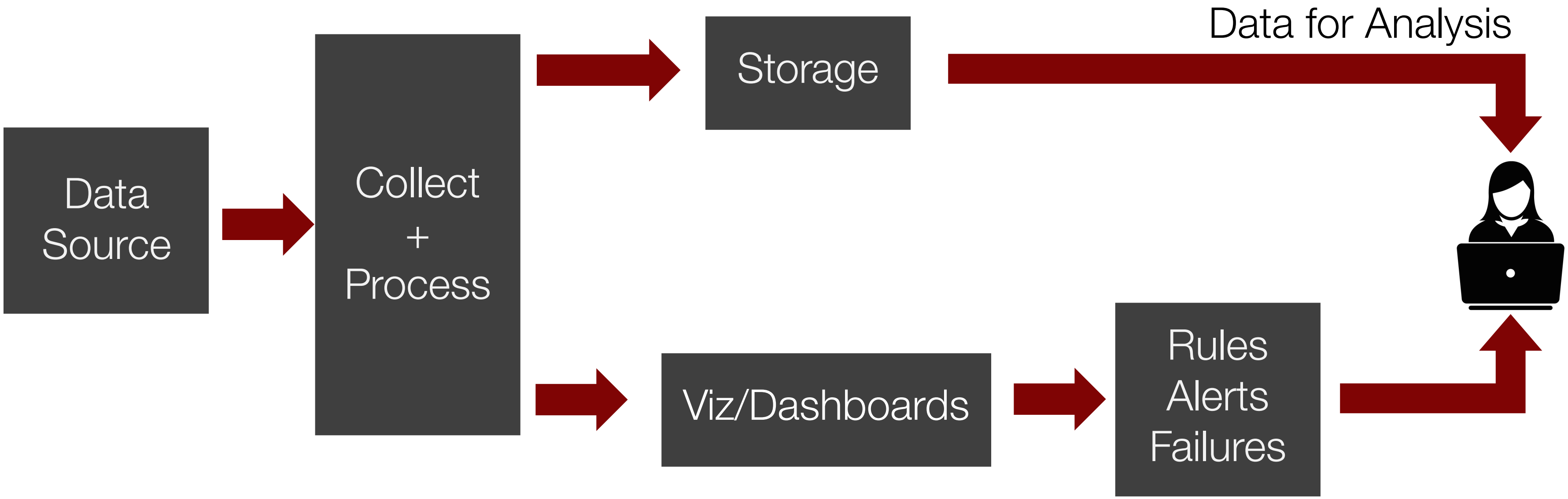
Prometheus

APACHE nifi

InfluxDB

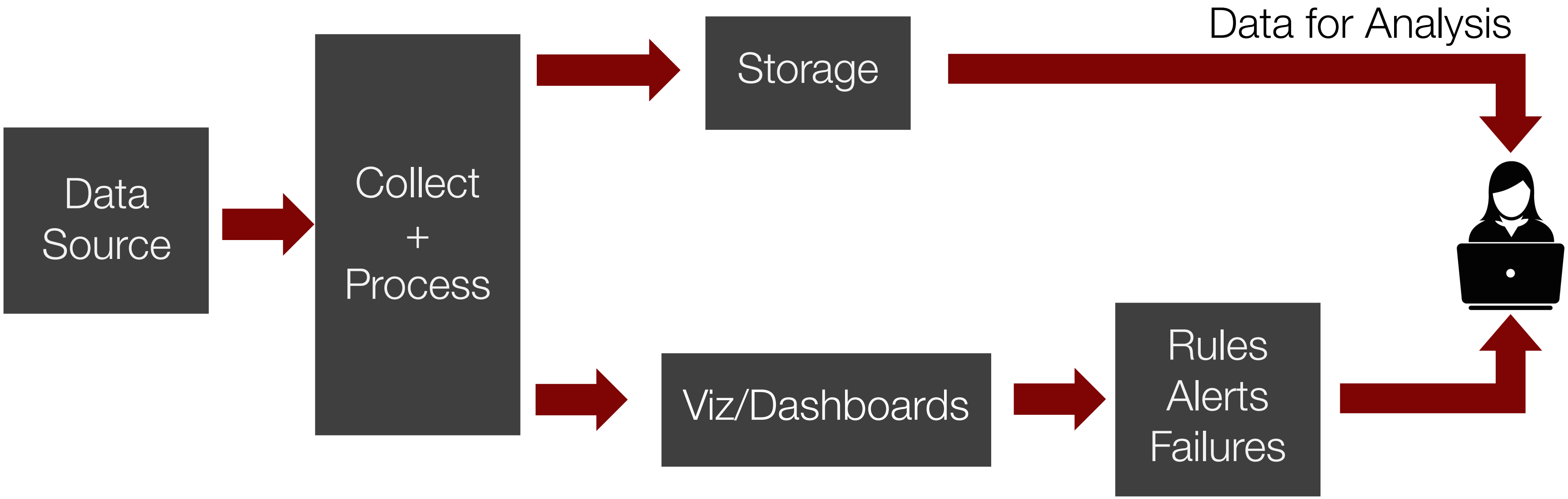
What’s the catch?

Typical Monitoring Pipeline



Typical Monitoring Pipeline

Top silicon valley orgs report using **less than 6%** of their data in this process.



Key Bottleneck: Human Attention

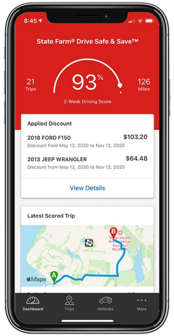
Human attention is limited!

MacroBase: A fast data analysis system

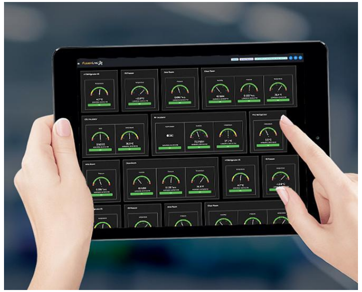
- It bridges the gap between the availability of low-level dataflow processing engines and the need for efficient, accurate analytics engines that prioritize attention in fast data

- It is a monitoring engine that prioritizes user attention by combining classification and explanation with streaming dataflow

Target Environments



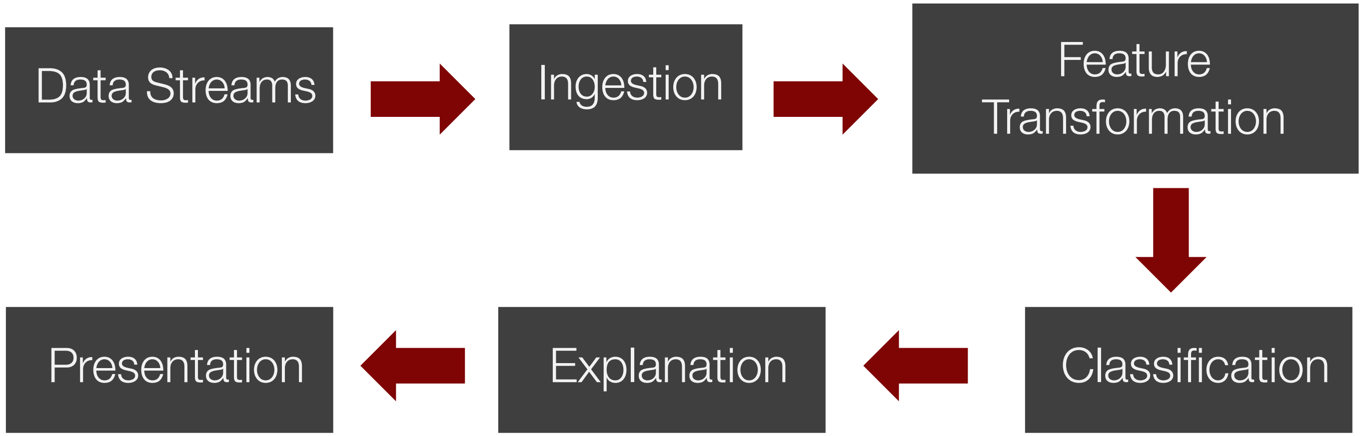
- **Mobile apps**: e.g., Cambridge Mobile Telematics App
- **Datacenter operation**: e.g., Amazon Web Services
- **Industrial monitoring**: e.g., temperature monitoring



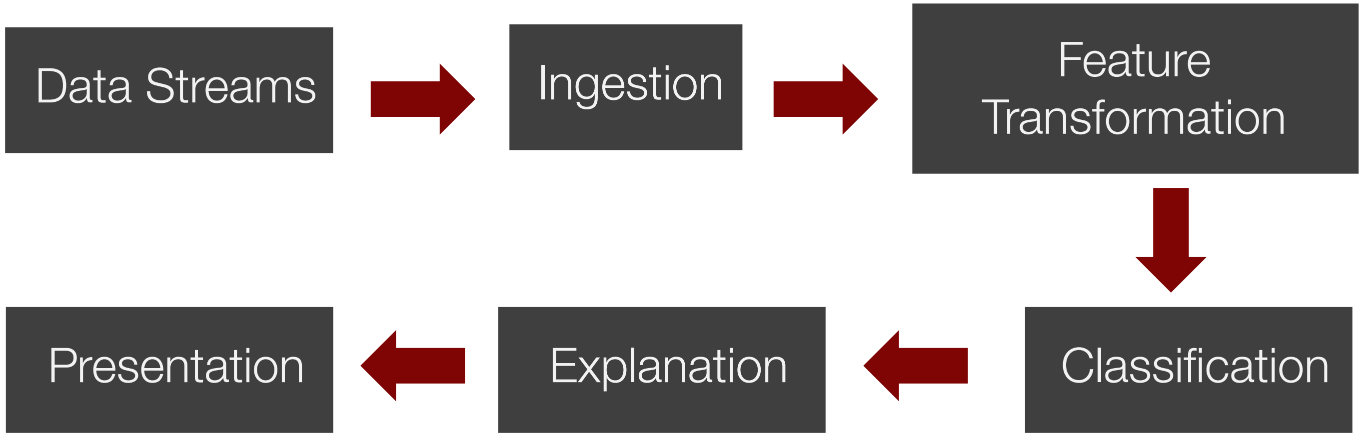
MacroBase Default Analysis Pipeline (MDP)

MacroBase executes **pipelines** of specialized dataflow operators over input data streams. Each MacroBase query specifies a set of input data sources as well as a logical query plan, or pipeline of streaming operators, that describes the analysis.

MacroBase Default Analysis Pipeline (MDP)



MacroBase Default Analysis Pipeline (MDP)

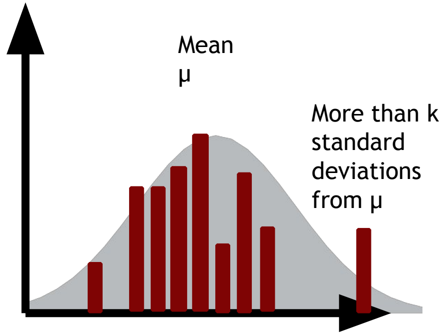


Extensibility: feature transformation, classification and explanation operators are all customizable

Ingestion and Feature Transformation

- **Ingestion**: MacroBase ingests data streams for analysis from a number of external data sources.
- **Feature Transformation**: MacroBase executes an optional series of domain-specific data transformations over the stream, which could include time-series specific operations, statistical operations, and datatype specific operations.

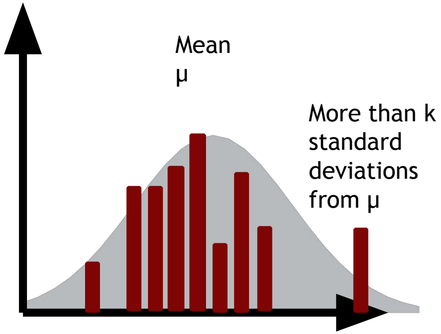
MDP Classification



- To classify is to **segment** and **filter** stream by target behavior (e.g., abnormalities)

- For example, Z score provides a normalized way to measure the “outlying”-ness of a point

MDP Classification



- MacroBase's classification operators use **unsupervised density-based classification** to label input data points and identify data points that exhibit deviant behavior
- Robust Estimators: **Median Absolute Deviation** (MAD) and **Minimum Covariance Determinant** (MCD)
- MacroBase uses **Adaptable Damped Reservoir** to retrain MAD and MCD while streaming

Ingestion → Feature Transformation → Classification → Explanation → Presentation

MDP Explanation

Errors

{iOS 9.0 beta 1, AT&T}
{iOS 9.0 beta 1, AT&T}
{iOS 8.8.5, T-Mobile}
{iOS 9.0 beta 1, T-Mobile}
{iOS 9.0 beta 1, AT&T}
{iOS 9.0 beta 1, T-Mobile}

Non-Errors

{iOS 9.0.1, AT&T}
{iOS 9.0.1, AT&T}
{iOS 9.0.1, T-Mobile}
{iOS 9.0.1, AT&T}
{iOS 8.8.5, T-Mobile}
{iOS 8.8.5, AT&T}
{iOS 9.0.1, T-Mobile}
{iOS 9.0.1, AT&T}

Maybe some problem with iOS 9.0 beta 1

Ingestion → Feature Transformation → Classification → Explanation → Presentation

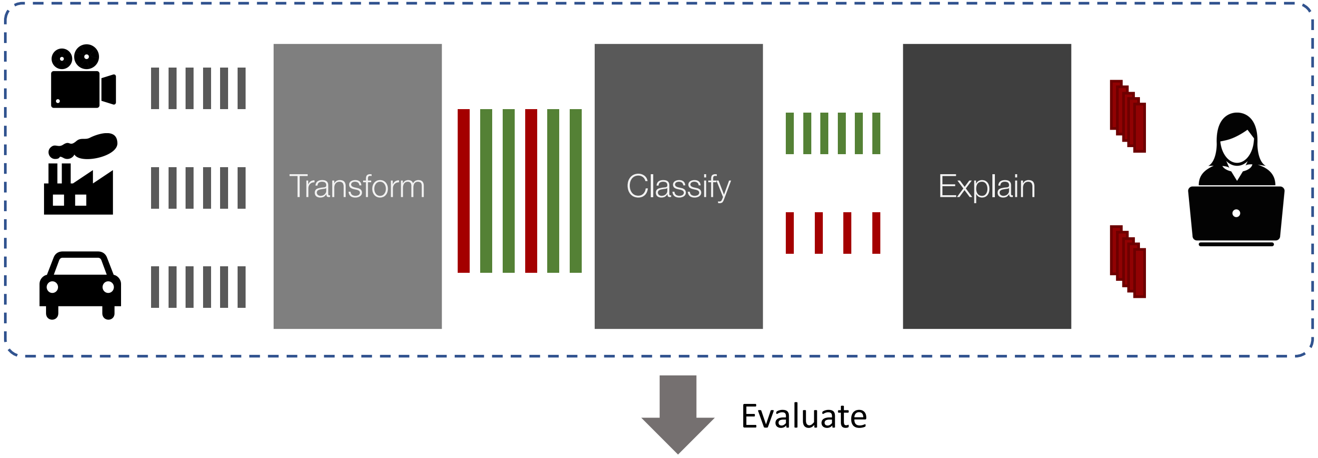
MDP Explanation

- MDP returns explanations in the form of **attribute-value combinations** (e.g., device ID 5052) that are common among outlier points but uncommon among inlier points.
- MDP’s streaming explanation operator utilizes an **Amortized Maintenance Counter** sketch and a prefix tree to maintain attributes in an approximate, frequency descending order.
- MacroBase periodically decays the counts of the items and the counts in each node of the prefix tree and prune any attributes that are no longer above the support threshold and rearranges the prefix tree in frequency-descending order.
- MacroBase runs FPGrowth on the prefix tree to produce explanations on demand.

Presentation

The number of output explanations may still be large. As a result, most pipelines rank explanations by statistics specific to the explanations before presentation.

EVALUATION



ACCURACY

- Synthetic datasets
- Real-world datasets

EFFICIENCY

- Adaptivity
- End-to-End Performance
- Cardinality Awareness
- AMC Comparison

FLEXIBILITY

- Hybrid Supervision
- Time-series
- Video Surveillance

EVALUATION ACCURACY EFFICIENCY FLEXIBILITY

Synthetic Dataset Accuracy

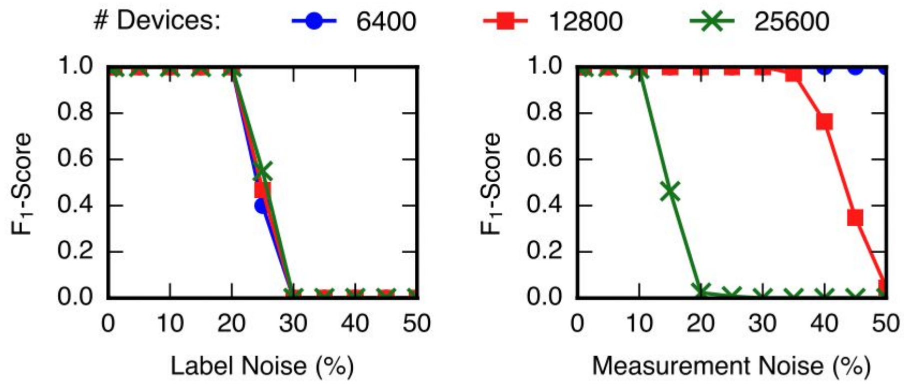


Fig. Precision-recall of explanations

- Recall that: $F_1\text{-score}\ \left(2 \cdot \frac{precision \cdot recall}{precision + recall}\right)$
- Dimensionality: 1M data points
- Each data point: device ID attribute and metrics drawn from either an inlier or outlier distribution
- Inlier distribution: N(10, 10)
- Outlier distribution: N(70, 10)

EVALUATION ACCURACY EFFICIENCY FLEXIBILITY

Synthetic Dataset Accuracy

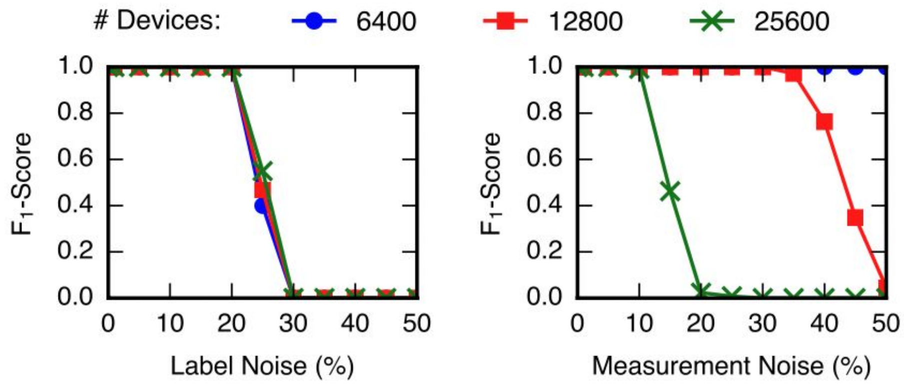


Fig. Precision-recall of explanations

Two types of Noise:

Label noise:

Randomly exchange readings of the outliers with inliers

Measurement noise:

Randomly assign a proportion of both outlying and inlying points to a third, uniform distribution over the interval [0,80]

- Recall that: $F_1\text{-score}\left(2 \cdot \frac{precision \cdot recall}{precision + recall}\right)$
- Dimensionality: 1M data points
- Each data point: device ID attribute and metrics drawn from either an inlier or outlier distribution
- Inlier distribution: N(10, 10)
- Outlier distribution: N(70, 10)

Synthetic Dataset Accuracy

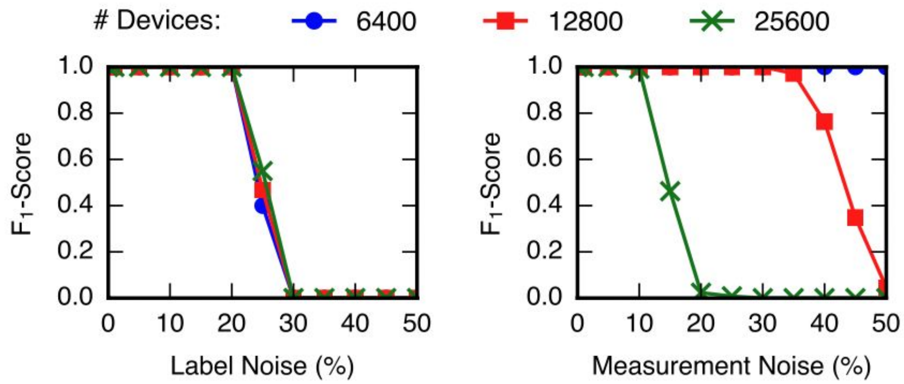


Fig. Precision-recall of explanations

Two types of Noise:

Label noise:

Randomly exchange readings of the outliers with inliers

Measurement noise:

Randomly assign a proportion of both outlying and inlying points to a third, uniform distribution over the interval [0,80]

- Recall that: $F_1\text{-score}\ \left(2 \cdot \frac{precision \cdot recall}{precision + recall}\right)$
- Dimensionality: 1M data points
- Each data point: device ID attribute and metrics drawn from either an inlier or outlier distribution
- Inlier distribution: N(10, 10)
- Outlier distribution: N(70, 10)

Key Observations:

- Under label noise, MacroBase robustly identified the outlying devices until 25% noise;

Synthetic Dataset Accuracy

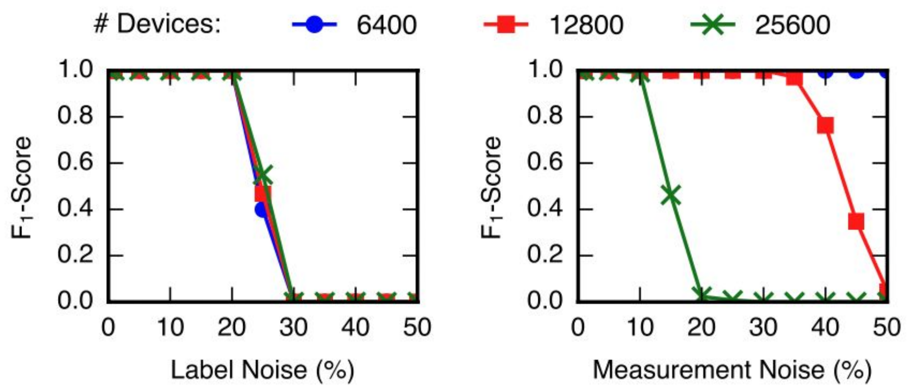


Fig. Precision-recall of explanations

Two types of Noise:

Label noise:

Randomly exchange readings of the outliers with inliers

Measurement noise:

Randomly assign a proportion of both outlying and inlying points to a third, uniform distribution over the interval [0,80]

- Recall that: $F_1\text{-score}\left(2 \cdot \frac{precision \cdot recall}{precision + recall}\right)$
- Dimensionality: 1M data points
- Each data point: device ID attribute and metrics drawn from either an inlier or outlier distribution
- Inlier distribution: N(10, 10)
- Outlier distribution: N(70, 10)

Key Observations:

- Under label noise, MacroBase robustly identified the outlying devices until 25% noise;

varying $\text{risk ratio} = \frac{a_o/(a_o+a_i)}{b_o/(b_o+b_i)}$ unchanged

Performance degraded when we exceed risk ratio threshold!

Synthetic Dataset Accuracy

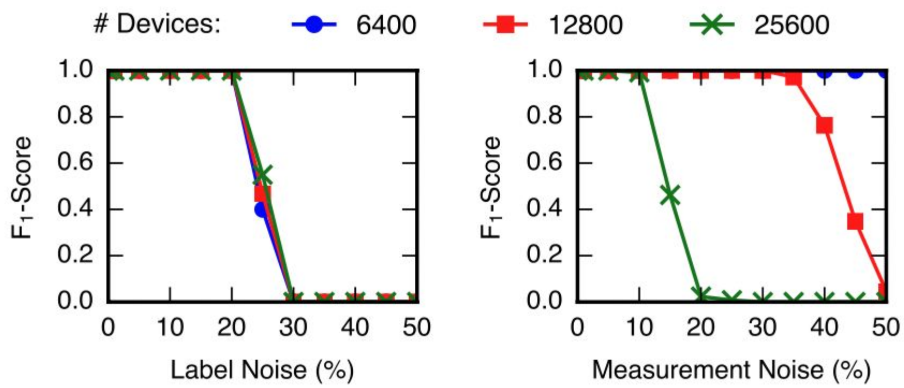


Fig. Precision-recall of explanations

Two types of Noise:

Label noise:

Randomly exchange readings of the outliers with inliers

Measurement noise:

Randomly assign a proportion of both outlying and inlying points to a third, uniform distribution over the interval [0,80]

- Recall that: $F_1\text{-score}\left(2 \cdot \frac{precision \cdot recall}{precision + recall}\right)$
- Dimensionality: 1M data points
- Each data point: device ID attribute and metrics drawn from either an inlier or outlier distribution
- Inlier distribution: N(10, 10)
- Outlier distribution: N(70, 10)

Key Observations:

- Under label noise, MacroBase robustly identified the outlying devices until 25% noise;
- Under measurement noise, accuracy degrades linearly with the amount of noise.

Real-world Dataset Accuracy

Distinguish abnormally-behaving in online transaction processing (OLTP) system.

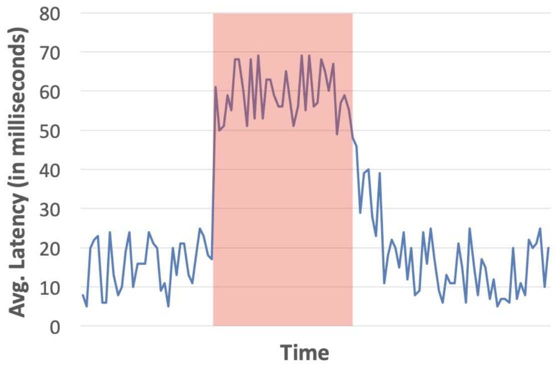


Fig. Example of a typical OLTP anomaly (latency)

- Experiments on performance degradation within MySQL on a particular OLTP workload (TPC-C and TPC-E).
- TPC-C (Transaction Processing Performance Council Benchmark C) & TPC-E (Transaction Processing Performance Council Benchmark E)

EVALUATION | **ACCURACY** | EFFICIENCY | FLEXIBILITY

Real-world Dataset Accuracy

	A1	A2	A3	A4	A5	A6	A7	A8	A9
TPC-C (QS: one MacroBase query per cluster): top-1: 88.8%, top-3: 88.8%									
Train top-1 correct (of 9)	9	9	9	9	9	8	9	9	8
Holdout top-1 correct (of 2)	2	2	2	2	2	2	2	2	0
TPC-C (QE: one MacroBase query per anomaly type): top-1: 83.3%, top-3: 100%									
Train top-1 correct (of 9)	9	9	9	9	9	8	9	9	7
Holdout top-1 correct (of 2)	2	2	2	2	2	1	2	2	0
TPC-E (QS: one MacroBase query per cluster): top-1: 83.3%, top-3: 88.8%									
Train top-1 correct (of 9)	9	9	9	9	9	8	9	9	0
Holdout top-1 correct (of 2)	2	2	2	2	2	1	2	2	0
TPC-E (QE: one MacroBase query per anomaly type): top-1: 94.4%, top-3: 100%									
Train top-1 correct (of 9)	9	9	9	9	9	8	9	9	6
Holdout top-1 correct (of 2)	2	2	2	2	2	1	2	2	2

Tab. MDP accuracy on DBSherlock workload.

A1: workload spike,
A2: I/O stress,
A3: DB backup,
A4: table restore,
A5: CPU stress,
A6: flush log/table;
A7: network congestion;
A8: lock contention;
A9: poorly written query.

QS

Metrics of A9 is different.

Real-world Dataset Accuracy

	A1	A2	A3	A4	A5	A6	A7	A8	A9
TPC-C (QS: one MacroBase query per cluster): top-1: 88.8%, top-3: 88.8%									
Train top-1 correct (of 9)	9	9	9	9	9	8	9	9	8
Holdout top-1 correct (of 2)	2	2	2	2	2	2	2	2	0
TPC-C (QE: one MacroBase query per anomaly type): top-1: 83.3%, top-3: 100%									
Train top-1 correct (of 9)	9	9	9	9	9	8	9	9	7
Holdout top-1 correct (of 2)	2	2	2	2	2	1	2	2	0
TPC-E (QS: one MacroBase query per cluster): top-1: 83.3%, top-3: 88.8%									
Train top-1 correct (of 9)	9	9	9	9	9	8	9	9	0
Holdout top-1 correct (of 2)	2	2	2	2	2	1	2	2	0
TPC-E (QE: one MacroBase query per anomaly type): top-1: 94.4%, top-3: 100%									
Train top-1 correct (of 9)	9	9	9	9	9	8	9	9	6
Holdout top-1 correct (of 2)	2	2	2	2	2	1	2	2	2

Tab. MDP accuracy on DBSherlock workload.

A1: workload spike,
A2: I/O stress,
A3: DB backup,
A4: table restore,
A5: CPU stress,
A6: flush log/table;
A7: network congestion;
A8: lock contention;
A9: poorly written query.

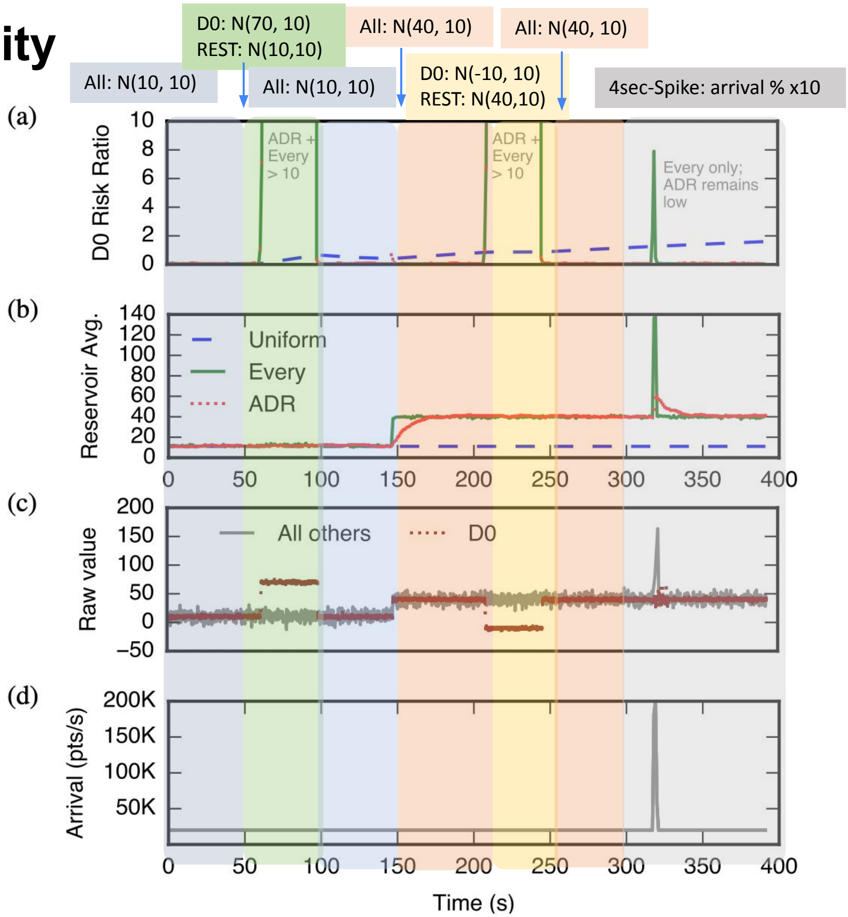
QS

QE

QE with higher Top-3 accuracy:
QE targets each type of performance degradation with a custom set of metrics.

With proper feature selection, MacroBase accurately recovers systemic causes even in unsupervised settings!

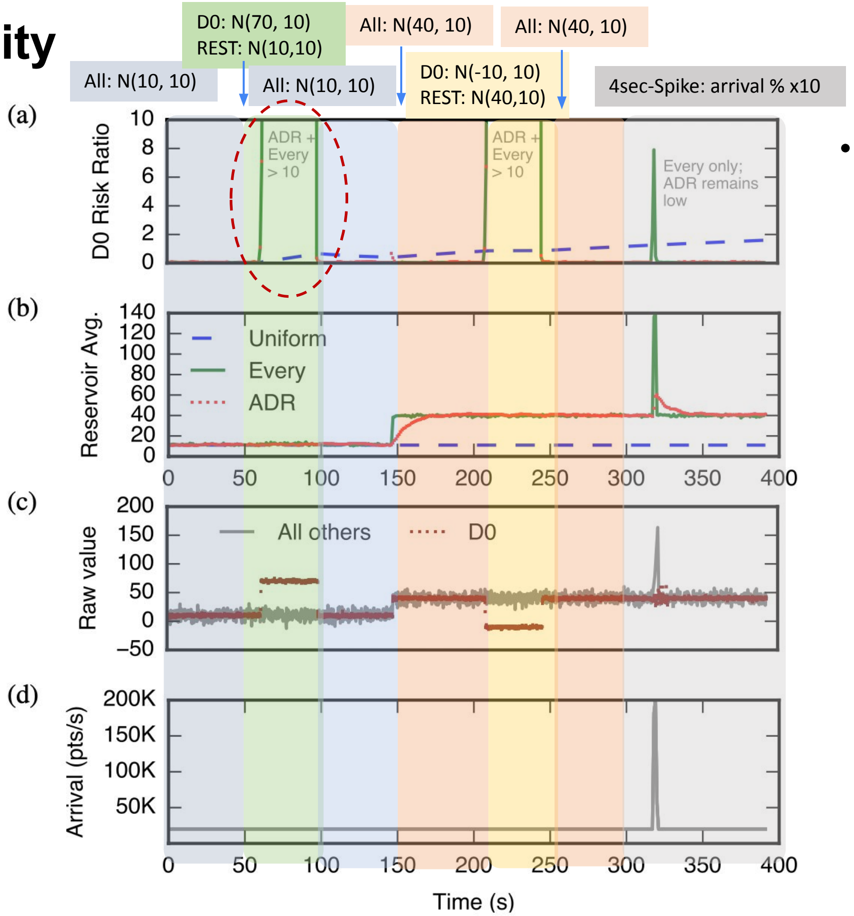
Adaptivity



Three sampling techniques:

- uniform reservoir sampling (Uniform)
- per-tuple exponentially decaying reservoir sampling (Every)
- ADR

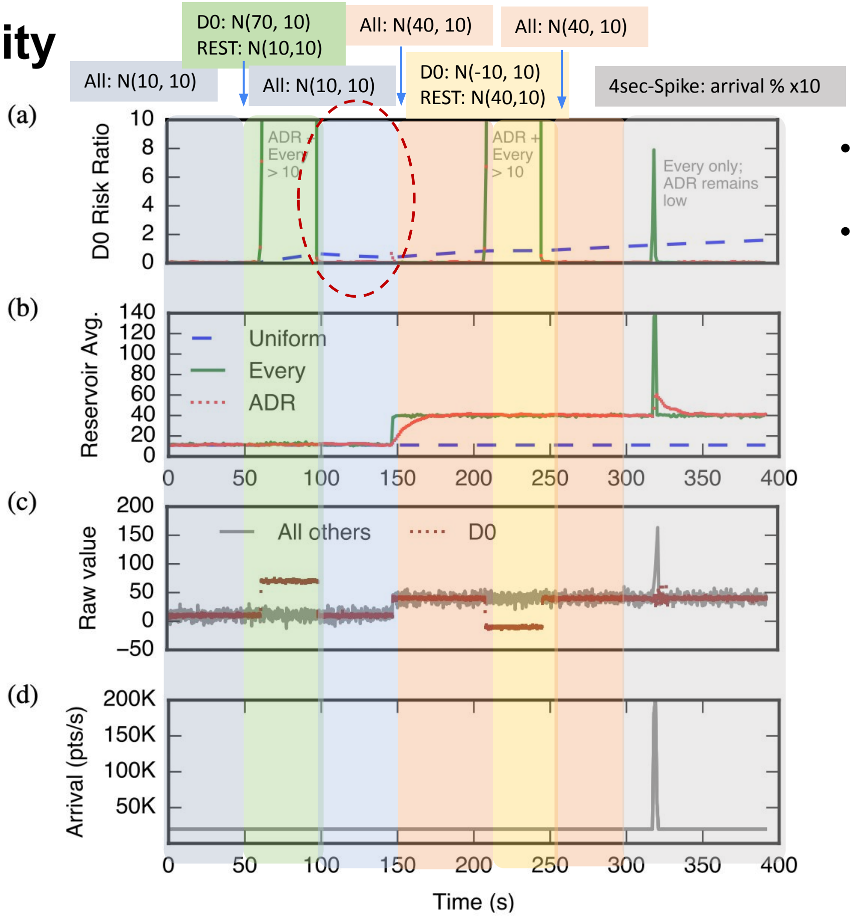
Adaptivity



- In the first period, all three methods detect D0 as an outlier;

EVALUATION ACCURACY EFFICIENCY FLEXIBILITY

Adaptivity



- In the first period, all three methods detect D0 as an outlier;
- In the second period, adaptive methods detect D0;

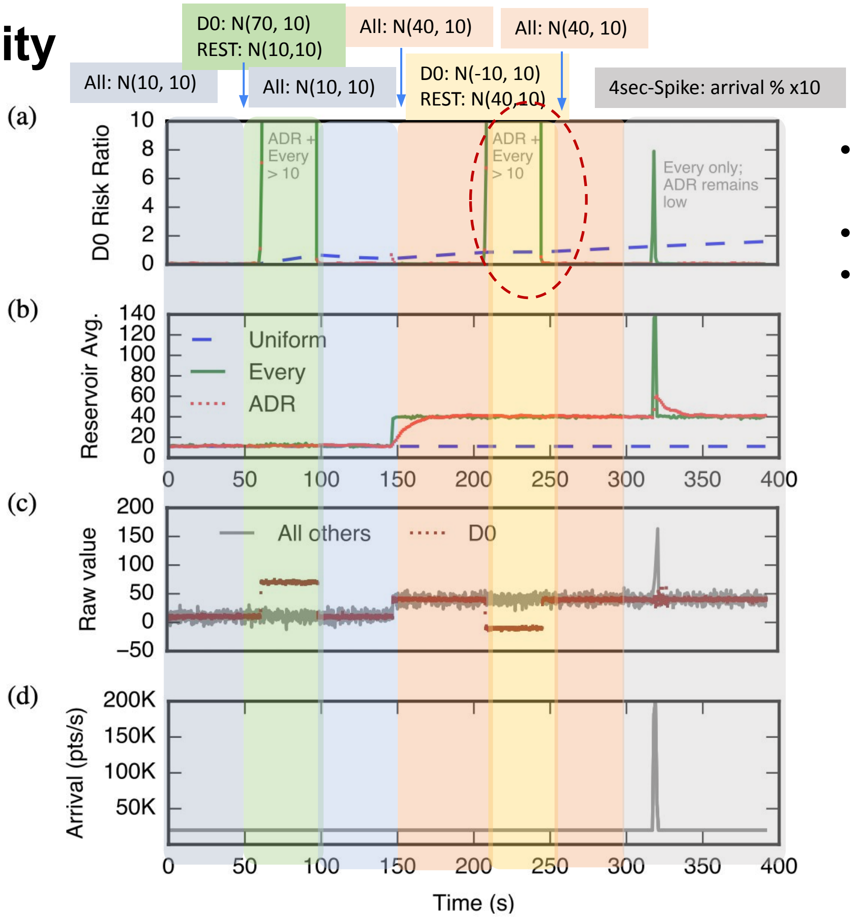
EVALUATION

ACCURACY

EFFICIENCY

FLEXIBILITY

Adaptivity

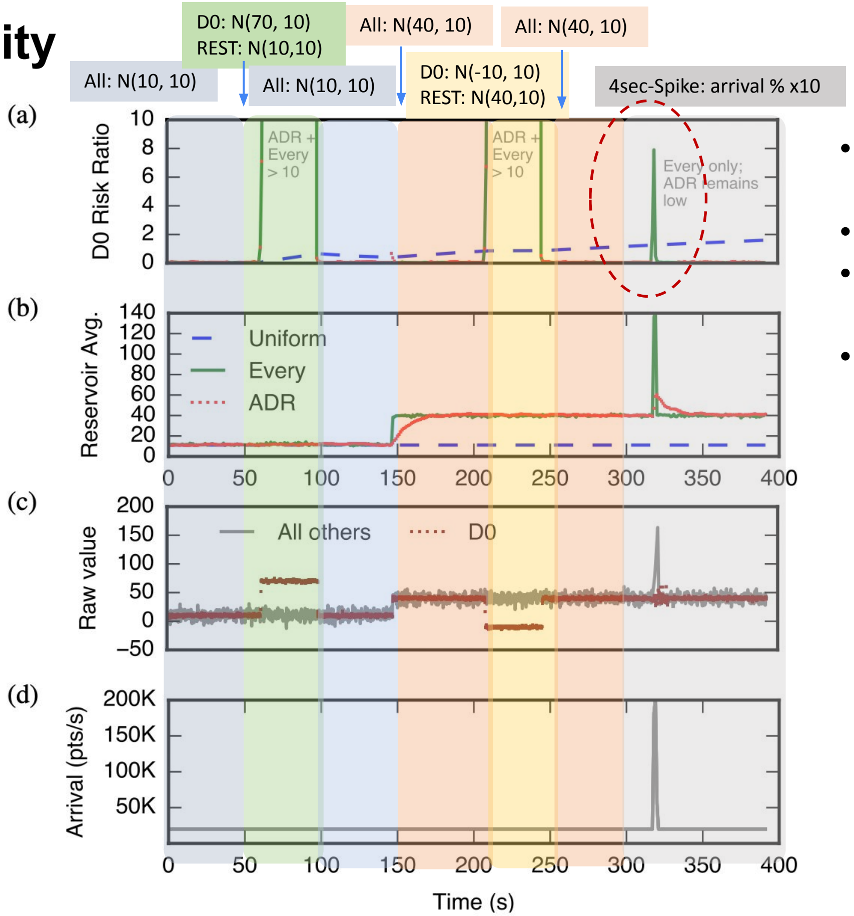


- In the first period, all three methods detect D0 as an outlier;
- In the second period, adaptive methods detect D0;
- In 225-250s, adaptive methods track the changes in D0;

EVALUATION

ACCURACY | EFFICIENCY | FLEXIBILITY

Adaptivity



- In the first period, all three methods detect D0 as an outlier;
- In the second period, adaptive methods detect D0;
- In 225-250s, adaptive methods track the changes in D0;
- At 300s, ADR is robust against the spike change; per-tuple method is subject to absorbing the spikes.

Adaptivity to distribution changes and resilience to variable arrival rates!

End-to-End Performance

Dataset	Queries Name	Metrics	Attrs	Points	Thru w/o Explain (pts/s) One-shot	EWS	Thru w/ Explain (pts/s) One-shot	EWS	# Explanations One-shot	EWS	Jaccard Similarity
Liquor	LS	1	1	3.05M	1549.7K	967.6K	1053.3K	966.5K	28	33	0.74
	LC	2	4		385.9K	504.5K	270.3K	500.9K	500	334	0.35
Telecom	TS	1	1	10M	2317.9K	698.5K	360.7K	698.0K	469	1	0.00
	TC	5	2		208.2K	380.9K	178.3K	380.8K	675	1	0.00
Campaign	ES	1	1	10M	2579.0K	778.8K	1784.6K	778.6K	2	2	0.67
	EC	1	5		2426.9K	252.5K	618.5K	252.1K	22	19	0.17
Accidents	AS	1	1	430K	998.1K	786.0K	729.8K	784.3K	2	2	1.00
	AC	3	3		349.9K	417.8K	259.0K	413.4K	25	20	0.55
Disburse	FS	1	1	3.48M	1879.6K	1209.9K	1325.8K	1207.8K	41	38	0.84
	FC	1	6		1843.4K	346.7K	565.3K	344.9K	1710	153	0.05
CMT	MS	1	1	10M	1958.6K	564.7K	354.7K	562.6K	46	53	0.63
	MC	7	6		182.6K	278.3K	147.9K	278.1K	255	98	0.29

Tab. Datasets and query names, throughput, and explanations produced under one-shot and exponentially weighted streaming (EWS) execution.

For each dataset X:

- XS: simple query w/ a single attribute and metric
- XC: complex query w/ more attributes and metrics (if applied)

Two system configs:

- One-shot batch execution: each stage is processed in sequence (examine the whole datasets at once)
- Exponentially-weighted streaming execution (EWS): points are processed continuously (focus on recent points)

EVALUATION ACCURACY **EFFICIENCY** FLEXIBILITY

End-to-End Performance

Dataset	Queries: Name	Metrics	Attrs	Points	Thru w/o Explain (pts/s): One-shot	EWS	Thru w/ Explain (pts/s): One-shot	EWS	# Explanations: One-shot	EWS	Jaccard Similarity
Liquor	LS	1	1	3.05M	1549.7K	967.6K	1053.3K	966.5K	28	33	0.74
	LC	2	4		385.9K	504.5K	270.3K	500.9K	500	334	0.35
Telecom	TS	1	1	10M	2317.9K	698.5K	360.7K	698.0K	469	1	0.00
	TC	5	2		208.2K	380.9K	178.3K	380.8K	675	1	0.00
Campaign	ES	1	1	10M	2579.0K	778.8K	1784.6K	778.6K	2	2	0.67
	EC	1	5		2426.9K	252.5K	618.5K	252.1K	22	19	0.17
Accidents	AS	1	1	430K	998.1K	786.0K	729.8K	784.3K	2	2	1.00
	AC	3	3		349.9K	417.8K	259.0K	413.4K	25	20	0.55
Disburse	FS	1	1	3.48M	1879.6K	1209.9K	1325.8K	1207.8K	41	38	0.84
	FC	1	6		1843.4K	346.7K	565.3K	344.9K	1710	153	0.05
CMT	MS	1	1	10M	1958.6K	564.7K	354.7K	562.6K	46	53	0.63
	MC	7	6		182.6K	278.3K	147.9K	278.1K	255	98	0.29

Tab. Datasets and query names, throughput, and explanations produced under one-shot and exponentially weighted streaming (EWS) execution.

Ave. Thru:

- One-shot: 1.39M
- EWS: 599K

Overall, one-shot generates more thru. than EWS, but it still depends heavily on the specific dataset and characteristics.

For each dataset X:

- XS: simple query w/ a single attribute and metric
- XC: complex query w/ more attributes and metrics (if applied)

Two system configs:

- One-shot batch execution: each stage is processed in sequence (examine the whole datasets at once)
- Exponentially-weighted streaming execution (EWS): points are processed continuously (focus on recent points)

EVALUATION ACCURACY **EFFICIENCY** FLEXIBILITY

End-to-End Performance

Dataset	Queries Name	Metrics	Attrs	Points	Thru w/o Explain (pts/s) One-shot	EWS	Thru w/ Explain (pts/s) One-shot	EWS	# Explanations One-shot	EWS	Jaccard Similarity
Liquor	LS	1	1	3.05M	1549.7K	967.6K	1053.3K	966.5K	28	33	0.74
	LC	2	4		385.9K	504.5K	270.3K	500.9K	500	334	0.35
Telecom	TS	1	1	10M	2317.9K	698.5K	360.7K	698.0K	469	1	0.00
	TC	5	2		208.2K	380.9K	178.3K	380.8K	675	1	0.00
Campaign	ES	1	1	10M	2579.0K	778.8K	1784.6K	778.6K	2	2	0.67
	EC	1	5		2426.9K	252.5K	618.5K	252.1K	22	19	0.17
Accidents	AS	1	1	430K	998.1K	786.0K	729.8K	784.3K	2	2	1.00
	AC	3	3		349.9K	417.8K	259.0K	413.4K	25	20	0.55
Disburse	FS	1	1	3.48M	1879.6K	1209.9K	1325.8K	1207.8K	41	38	0.84
	FC	1	6		1843.4K	346.7K	565.3K	344.9K	1710	153	0.05
CMT	MS	1	1	10M	1958.6K	564.7K	354.7K	562.6K	46	53	0.63
	MC	7	6		182.6K	278.3K	147.9K	278.1K	255	98	0.29

Tab. Datasets and query names, throughput, and explanations produced under one-shot and exponentially weighted streaming (EWS) execution.

Generally, queries with multiple metrics in one-shot are slower than queries with single metrics, due to increased training time, as streaming trains over samples.

For each dataset X:

- XS: simple query w/ a single attribute and metric
- XC: complex query w/ more attributes and metrics (if applied)

Two system configs:

- One-shot batch execution: each stage is processed in sequence (examine the whole datasets at once)
- Exponentially-weighted streaming execution (EWS): points are processed continuously (focus on recent points)

End-to-End Performance

Dataset	Queries Name	Metrics	Attrs	Points	Thru w/o Explain (pts/s) One-shot	EWS	Thru w/ Explain (pts/s) One-shot	EWS	# Explanations One-shot	EWS	Jaccard Similarity
Liquor	LS	1	1	3.05M	1549.7K	967.6K	1053.3K	966.5K	28	33	0.74
	LC	2	4		385.9K	504.5K	270.3K	500.9K	500	334	0.35
Telecom	TS	1	1	10M	2317.9K	698.5K	360.7K	698.0K	469	1	0.00
	TC	5	2		208.2K	380.9K	178.3K	380.8K	675	1	0.00
Campaign	ES	1	1	10M	2579.0K	778.8K	1784.6K	778.6K	2	2	0.67
	EC	1	5		2426.9K	252.5K	618.5K	252.1K	22	19	0.17
Accidents	AS	1	1	430K	998.1K	786.0K	729.8K	784.3K	2	2	1.00
	AC	3	3		349.9K	417.8K	259.0K	413.4K	25	20	0.55
Disburse	FS	1	1	3.48M	1879.6K	1209.9K	1325.8K	1207.8K	41	38	0.84
	FC	1	6		1843.4K	346.7K	565.3K	344.9K	1710	153	0.05
CMT	MS	1	1	10M	1958.6K	564.7K	354.7K	562.6K	46	53	0.63
	MC	7	6		182.6K	278.3K	147.9K	278.1K	255	98	0.29

Tab. Datasets and query names, throughput, and explanations produced under one-shot and exponentially weighted streaming (EWS) execution.

For datasets with few distinct attribute values (Accidents contains only 9 types of weather conditions), the explanations will have high similarity. However, explanations differ in datasets with many distinct attribute values.

For each dataset X:

- XS: simple query w/ a single attribute and metric
- XC: complex query w/ more attributes and metrics (if applied)

Two system configs:

- One-shot batch execution: each stage is processed in sequence (examine the whole datasets at once)
- Exponentially-weighted streaming execution (EWS): points are processed continuously (focus on recent points)

EVALUATION ACCURACY **EFFICIENCY** FLEXIBILITY

End-to-End Performance

Dataset	Queries Name	Metrics	Attrs	Points	Thru w/o Explain (pts/s) One-shot	EWS	Thru w/ Explain (pts/s) One-shot	EWS	# Explanations One-shot	EWS	Jaccard Similarity
Liquor	LS	1	1	3.05M	1549.7K	967.6K	1053.3K	966.5K	28	33	0.74
	LC	2	4		385.9K	504.5K	270.3K	500.9K	500	334	0.35
Telecom	TS	1	1	10M	2317.9K	698.5K	360.7K	698.0K	469	1	0.00
	TC	5	2		208.2K	380.9K	178.3K	380.8K	675	1	0.00
Campaign	ES	1	1	10M	2579.0K	778.8K	1784.6K	778.6K	2	2	0.67
	EC	1	5		2426.9K	252.5K	618.5K	252.1K	22	19	0.17
Accidents	AS	1	1	430K	998.1K	786.0K	729.8K	784.3K	2	2	1.00
	AC	3	3		349.9K	417.8K	259.0K	413.4K	25	20	0.55
Disburse	FS	1	1	3.48M	1879.6K	1209.9K	1325.8K	1207.8K	41	38	0.84
	FC	1	6		1843.4K	346.7K	565.3K	344.9K	1710	153	0.05
CMT	MS	1	1	10M	1958.6K	564.7K	354.7K	562.6K	46	53	0.63
	MC	7	6		182.6K	278.3K	147.9K	278.1K	255	98	0.29

Tab. Datasets and query names, throughput, and explanations produced under one-shot and exponentially weighted streaming (EWS) execution.

EWS generates fewer explanations than one-shot (temporal bias)

In practice, users tune their decay on a per-application basis.

For each dataset X:

- XS: simple query w/ a single attribute and metric
- XC: complex query w/ more attributes and metrics (if applied)

Two system configs:

- One-shot batch execution: each stage is processed in sequence (examine the whole datasets at once)
- Exponentially-weighted streaming execution (EWS): points are processed continuously (focus on recent points)

EVALUATION ACCURACY **EFFICIENCY** FLEXIBILITY

End-to-End Performance

Dataset	Queries Name	Queries Metrics	Queries Attrs	Points	Thru w/o Explain (pts/s) One-shot	Thru w/o Explain (pts/s) EWS	Thru w/ Explain (pts/s) One-shot	Thru w/ Explain (pts/s) EWS	# Explanations One-shot	# Explanations EWS	Jaccard Similarity
Liquor	LS	1	1	3.05M	1549.7K	967.6K	1053.3K	966.5K	28	33	0.74
	LC	2	4		385.9K	504.5K	270.3K	500.9K	500	334	0.35
Telecom	TS	1	1	10M	2317.9K	698.5K	360.7K	698.0K	469	1	0.00
	TC	5	2		208.2K	380.9K	178.3K	380.8K	675	1	0.00
Campaign	ES	1	1	10M	2579.0K	778.8K	1784.6K	778.6K	2	2	0.67
	EC	1	5		2426.9K	252.5K	618.5K	252.1K	22	19	0.17
Accidents	AS	1	1	430K	998.1K	786.0K	729.8K	784.3K	2	2	1.00
	AC	3	3		349.9K	417.8K	259.0K	413.4K	25	20	0.55
Disburse	FS	1	1	3.48M	1879.6K	1209.9K	1325.8K	1207.8K	41	38	0.84
	FC	1	6		1843.4K	346.7K	565.3K	344.9K	1710	153	0.05
CMT	MS	1	1	10M	1958.6K	564.7K	354.7K	562.6K	46	53	0.63
	MC	7	6		182.6K	278.3K	147.9K	278.1K	255	98	0.29

Tab. Datasets and query names, throughput, and explanations produced under one-shot and exponentially weighted streaming (EWS) execution.

Runtime Contributions:
- MS: MAD (54%)
- MC: MCD (52%)
- FC: Gen-EXP (65%)

The overhead of each component is data- and query-dependent.

For each dataset X:
- XS: simple query w/ a single attribute and metric
- XC: complex query w/ more attributes and metrics (if applied)

Two system configs:
- One-shot batch execution: each stage is processed in sequence (examine the whole datasets at once)
- Exponentially-weighted streaming execution (EWS): points are processed continuously (focus on recent points)

Cardinality Awareness

- MacroBase leverages a unique *pruning strategy (support and risk ratio)* that harnesses the low cardinality of outliers and thus leads to large *speedups*;
- MacroBase produces a summary of each dataset's inliers and outliers in 0.22–1.4 seconds, i.e., 3.2x faster than unoptimized FPGrowth.
- Both are lower bounded by the linear pass over all inliers.

$$\text{risk ratio} = \frac{a_o/(a_o + a_i)}{b_o/(b_o + b_i)}$$

Cardinality Awareness

- MacroBase leverages a unique *pruning strategy (support and risk ratio)* that harnesses the low cardinality of outliers and thus leads to large *speedups*;
- MacroBase produces a summary of each dataset's inliers and outliers in 0.22–1.4 seconds, i.e., 3.2x faster than unoptimized FPGrowth.
- Both are lower bounded by the linear pass over all inliers.

AMC Comparison

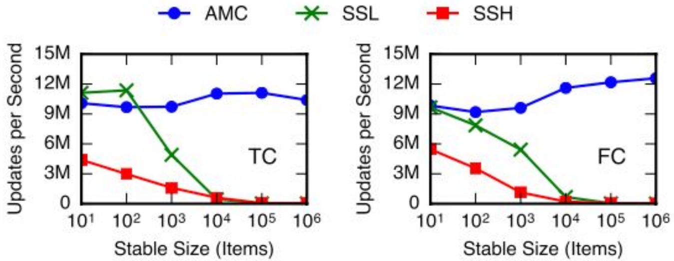


Fig. Streaming heavy hitters sketch comparison

- AMC: Amortized Maintenance Counter;
- SSL: Space Saving List;
- SSH: Space Saving Hash.

- The Space Saving overhead is costly, because list traversal and heap maintenance on every operation is expensive.
- AMC trades space for performance.

When memory sizes are especially constrained, Space Saving may be preferable.

EVALUATION ACCURACY EFFICIENCY **FLEXIBILITY**

Hybrid Supervision: CMT

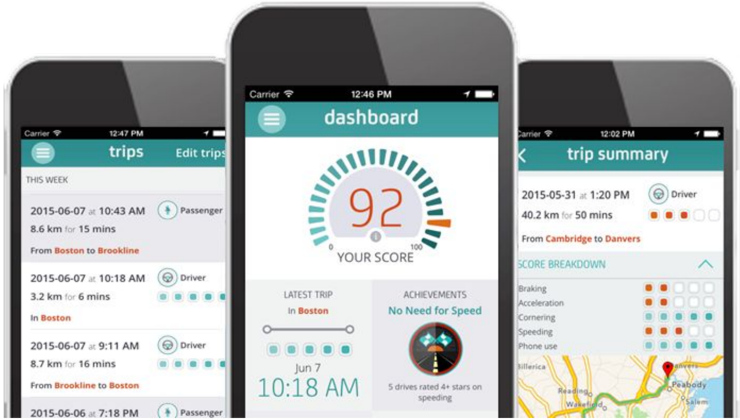
Cambridge Mobile Telematics:

Monitors driving behavior via mobile application available for smartphones

Question: Is the application behaving correctly on every platform?

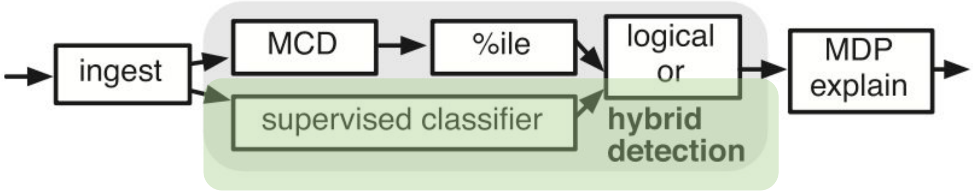
Extra Input: Each trip in the CMT dataset is accompanied by a supervised diagnostic score representing the trip quality.

Extra Target: We also want to capture anomalies with a low supervised diagnostic score. (independent of the primal distribution)



Hybrid Supervision: CMT

Pipeline variant #1

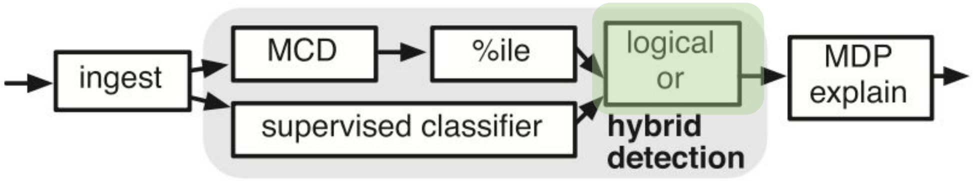


Add parallel path for supervised classification

* Supervised classifier: special rule-based operator that flags low quality scores as anomalies.

Hybrid Supervision: CMT

Pipeline variant #1



Use a logical OR gate for integrating results

Runtime Analysis

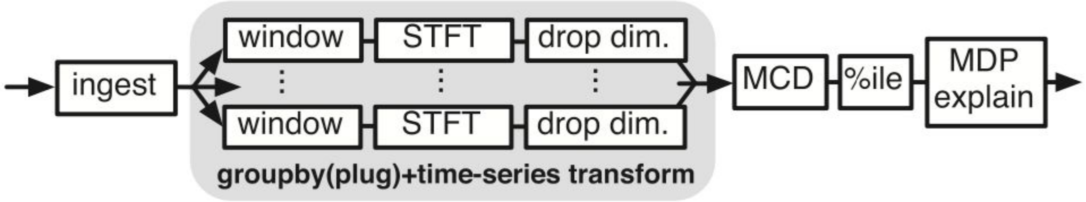
- Runtime remains unaffected since the external rule-based path is lightweight.

EVALUATION

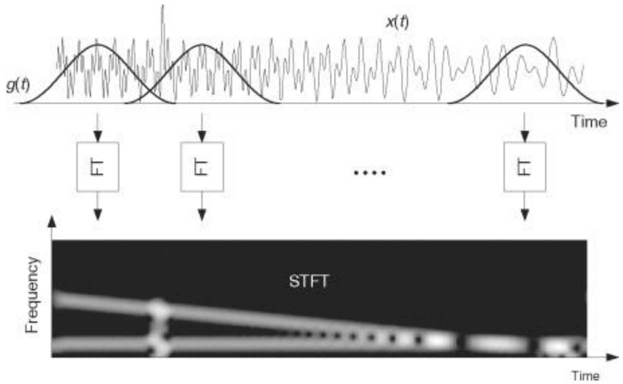
ACCURACY | EFFICIENCY | FLEXIBILITY

Time-series On dataset of 16M points capturing a month of electricity usage from devices within a household

Pipeline variant #2



Discrete-Time Short-Term Fourier Transform (STFT)



Window applied on the signal

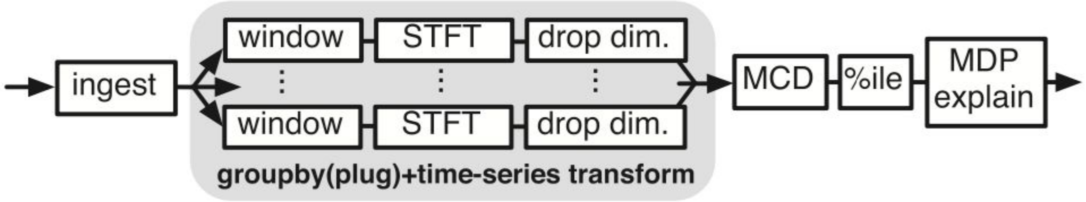
FT applied on each window

Frequency-time plot

Time-series

On dataset of 16M points capturing a month of electricity usage from devices within a household

Pipeline variant #2



Partition the stream by device ID

Window the stream into hourly intervals w\ attributes according to hour, day, week, and date

Apply a STFT to each window

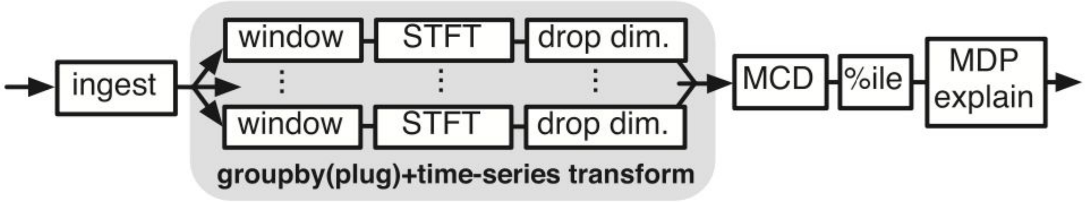
Truncate the transformed data to a fixed number of dimensions.

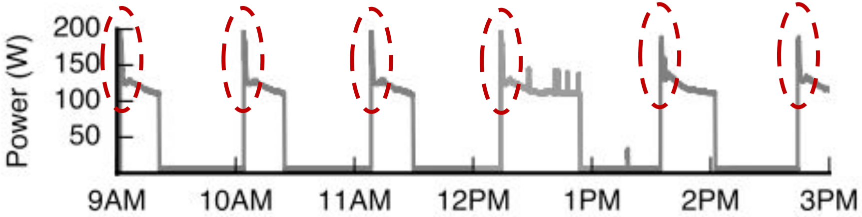
EVALUATION ACCURACY EFFICIENCY **FLEXIBILITY**

Time-series

On dataset of 16M points capturing a month of electricity usage from devices within a household

Pipeline variant #2





Spikes: do they indicate anomalies?

NO!

household refrigerator spiked on an hourly basis possibly corresponding to compressor activity

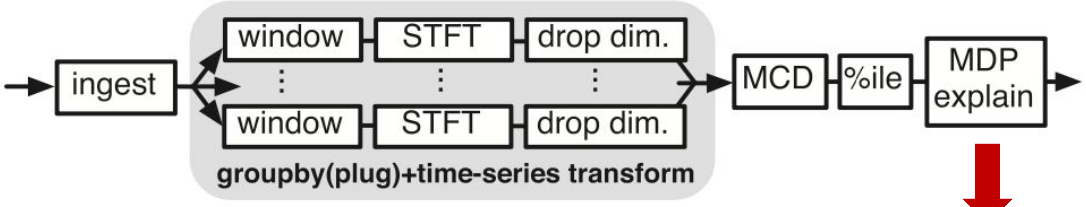
EVALUATION

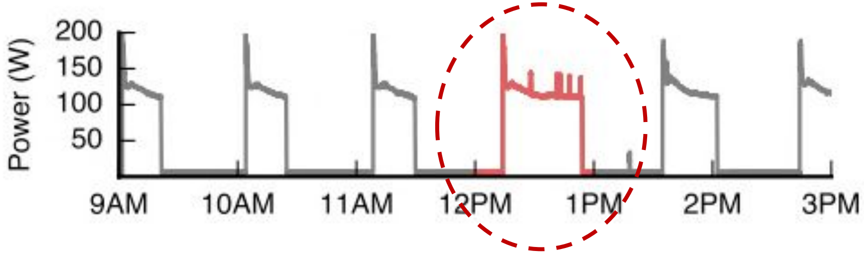
ACCURACY | EFFICIENCY | **FLEXIBILITY**

Time-series

On dataset of 16M points capturing a month of electricity usage from devices within a household

Pipeline variant #2





Real anomaly detected:

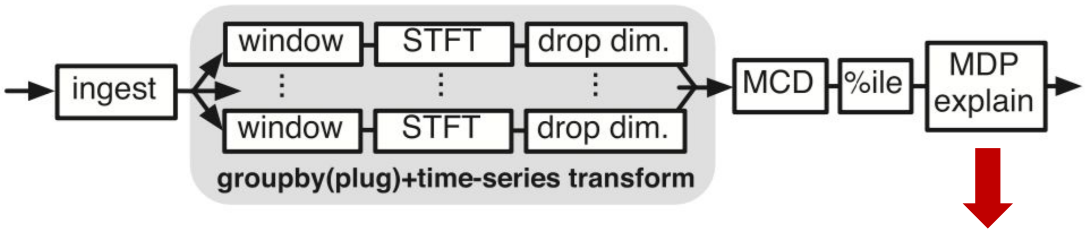
Refrigerator consistently behaved abnormally compared to other devices in the household and to other time periods between the hours of 12PM and 1PM.

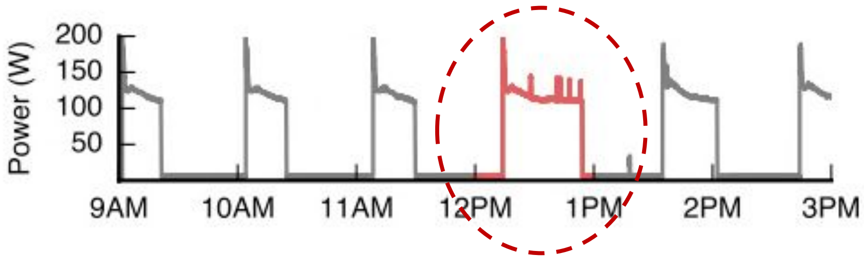
EVALUATION

ACCURACY | EFFICIENCY | **FLEXIBILITY**

Time-series On dataset of 16M points capturing a month of electricity usage from devices within a household

Pipeline variant #2





Real anomaly detected:

Refrigerator consistently behaved abnormally compared to other devices in the household and to other time periods between the hours of 12PM and 1PM.

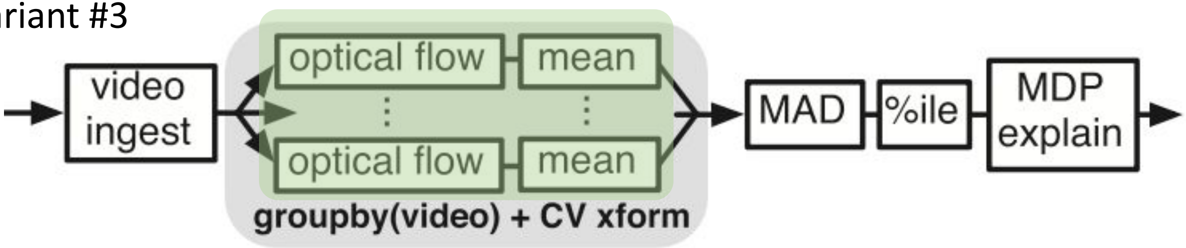
Runtime Analysis

- Without feature transformation, the entire pipeline was completed in 158ms.
- Feature transformation dominated the runtime, i.e., 516s to transform the 16M points via unoptimized STFT.

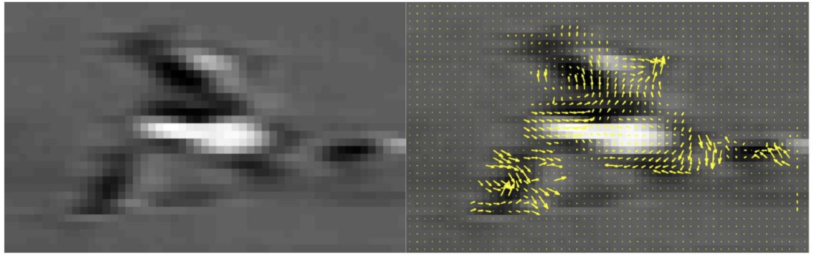
Video Surveillance

CAVIAR video surveillance dataset

Pipeline variant #3



Based on OpenCV, we add a custom feature transform that computes the average optical flow velocity between video frames



(a) original image (b) optical flow $F_{x,y}$

Video Surveillance

CAVIAR video surveillance dataset

Pipeline variant #3

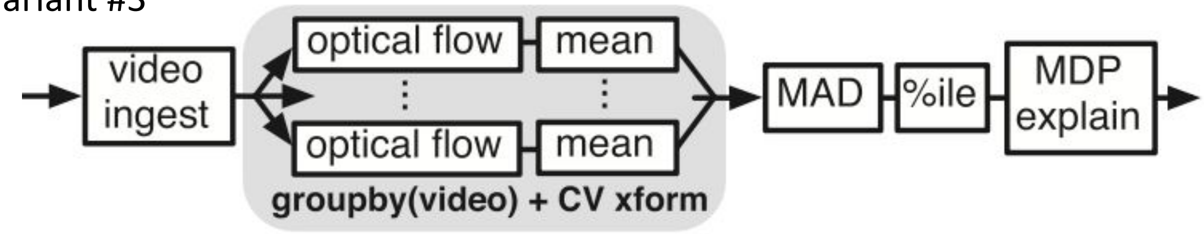


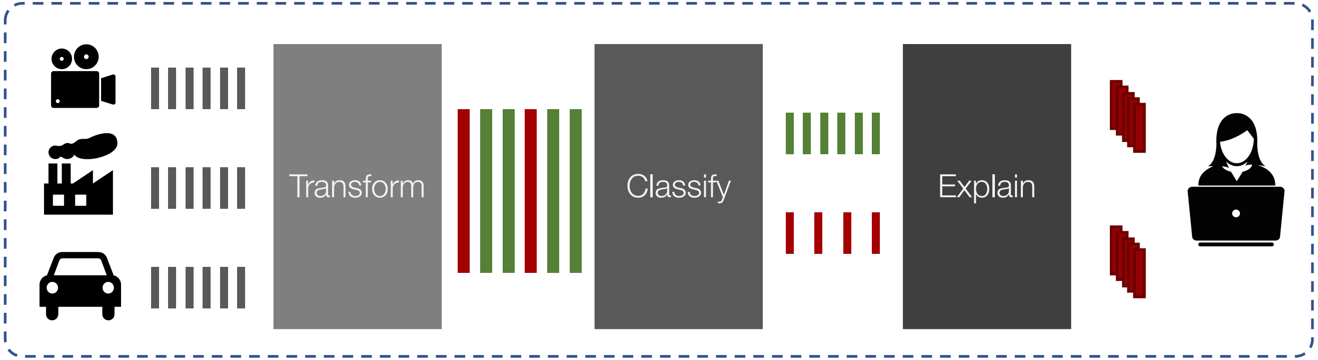
Fig. Frames containing violence highlighted by MDP

Runtime Analysis

- Feature transformation via optical flow dominated runtime (22s vs. 34ms for MDP);
  → adopted transform is expensive on CPU-based implementation

IMPLEMENTATION DETAILS

9,400 lines of Java, over 7,000 of which are devoted to operator implementation, along with an additional 1,000 lines of JS and HTML for the front-end and 7,600 lines of Java for diagnostics and prototype pipelines.



IMPLEMENTATION DETAILS

Implementation on Java:

+ high productivity
+ support for higher-order functions
+ popularity in open source

- performance overhead w/ the Java virtual machine (JVM)

	LS	TS	ES	AS	FS	MS
Throughput (points/sec)	7.86M	8.70M	9.35M	12.31M	7.05M	6.22M
Speedup over Java	7.46×	24.11×	5.24×	16.87×	5.32×	17.54×

Tab. Performance superiority of a simplified MDP pipeline in hand-optimized C++.

RELATED WORK

Streaming and Specialized Analytics

- Storm, StreamBase, IBM Oracle Streams (and so on) provide infrastructure for executing streaming queries
- MDP aims to provide a set of high-level analytic monitoring operators where dataflow is a means to an end rather than an end in itself
- Inspirations from specialized engines: Gigascope (network monitoring), WaveScope (signal processing), MCDB (Monte Carlo-based operators), and Bismarck (extensible aggregation for gradient-based optimization)
- Even though many commercially-available analytics packages provide advanced analytics functionality, none provides streaming explanation operations as in MacroBase.

RELATED WORK

Streaming and Specialized Analytics

- Storm, StreamBase, IBM Oracle Streams (and so on) provide infrastructure for executing streaming queries
- MDP aims to provide a set of high-level analytic monitoring operators where dataflow is a means to an end rather than an end in itself
- Inspirations from specialized engines: Gigascope (network monitoring), WaveScope (signal processing), MCDB (Monte Carlo-based operators), and Bismarck (extensible aggregation for gradient-based optimization)
- Even though many commercially-available analytics packages provide advanced analytics functionality, none provides streaming explanation operations as in MacroBase.

Classification

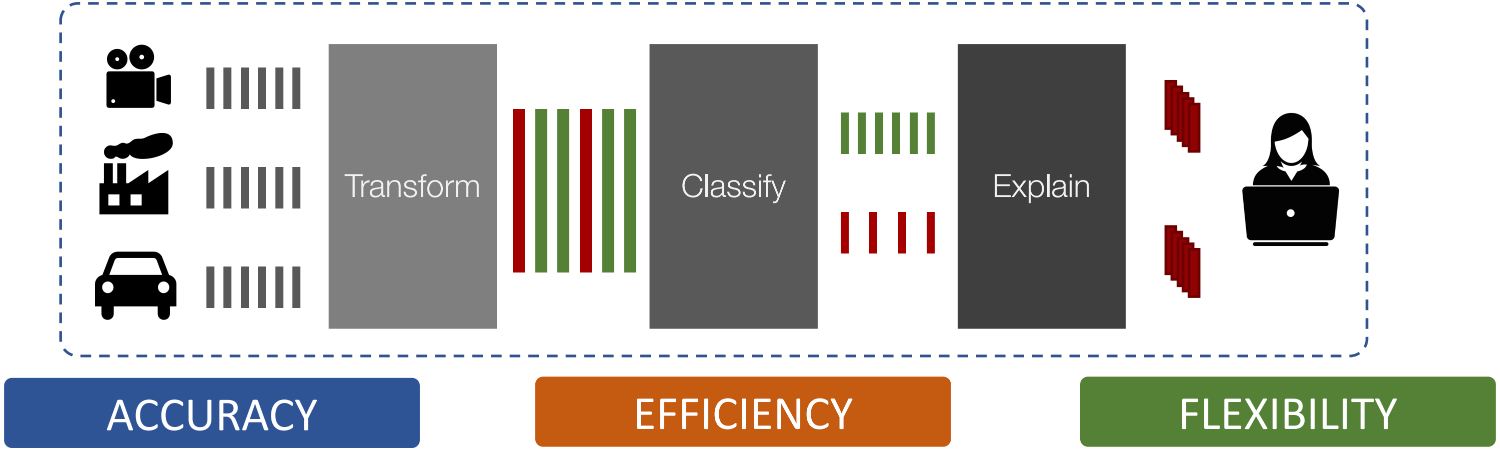
- A large amount of technique for classification and outlier detection emerging from statistics, machine learning, data mining, etc.
- Statistical outlier detection for stream volume techniques will produce a large stream of outlying data points, thus being coupled with streaming explanation.
- MacroBase is compatible with other detectors from Elki, Weka, RapidMiner, and OpenGamma.

RELATED WORK

Data Explanation

- Existing explanation techniques: decision-tree, Apriori-like pruning, grid search, data cubing, Bayesian statistics, visualization, causal reasoning, etc.
- none of the above techniques executes over streaming data or is efficient at the large dataset scale. Several exhibit runtime exponential in the number of attributes.
- MacroBase's explanation techniques take advantage of sketching and streaming data structures and adapt to the fast data setting, with compatibility to the existing explanation techniques.

CONCLUSION AND FUTURE WORK



MacroBase is

- a new analytics engine designed to prioritize attention in fast data streams
- a flexible architecture with streaming classification and data explanation to deliver interpretable summaries of important behaviors in fast data streams
- specifically optimized for high-volume, time-sensitive, and heterogeneous data streams

Future: new functionalities to expand domain supports and integrate more features by leveraging the flexibility provided by MacroBase's pipeline architecture.

THANK YOU!

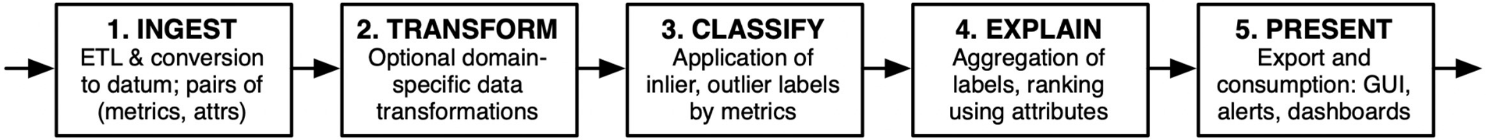
MacroBase: Prioritizing Attention in Fast Data

Researcher’s Perspective by: Cuong (Johnny) Nguyen

Summary of MacroBase

- First dataflow engine architecture to combine both streaming outlier detection and group-level explanation of outliers for time series anomaly detection by proposing SQL-like operators for these tasks

- Effective on large and fast dataflows, capable of processing several hundred thousands - 2.5 million data points per second on real datasets



Extensions of MacroBase

- Quantile estimation and explanation a common task for analysis of fast dataflow

- More fine-grained analysis between multiple classes of data points given a target variable, beyond the inlier vs outlier distinction presented in the paper

- **E.g:** Finding attributes that could explaining the difference in power consumption between devices in 75th percentile vs devices in the 90th percentile

Follow-up Research: MesoBase

- Adds a “quantile” operator to the MacroBase analytics pipeline, allows for accurate and timely estimation of quantiles for all data points given a target variable on fast data streams (see Quantiles on Streams, Buragohain and Suri 2009)

- Adapt the explanation framework presented in MacroBase for comparison between quantiles

- Evaluate the “quantile” operator on data streams similar to the evaluation method of MacroBase

MacroBase
A Peer Review

Reviewer: Eric Martin

Main Contributions

- First fast data stream analytics engine and pipeline architecture combining streaming outlier classification with streaming data explanation.
- A new type of exponentially damped reservoir sampler (Adaptable Damped Reservoir) which can operate of arbitrary window sizes.
- An optimization for calculating relative risk of outliers over inliers exploiting class imbalances between the two in fast data.
- A new heavy-hitters sketch which leverages more memory to provide quicker updates at higher accuracy. (Amortized Maintenance Counter)

Strong Points

- Leverages existing concepts in pipeline and data processing architectures which increasing the likelihood of adoption. **Hard to implement but easy to plug in.**
- Introduces a type system which...
  - Promote the importance of classification and explanation as data types.
  - Creates a extendable system where new operators can be added/optimized.
  - Production case studies are proofs of concept.
- Create new types of samplers, sketches, and optimizations based on the inputs and outputs of each operator. **Sum of the operators is greater than their parts.**
- Test the system with both simulated and real world data.

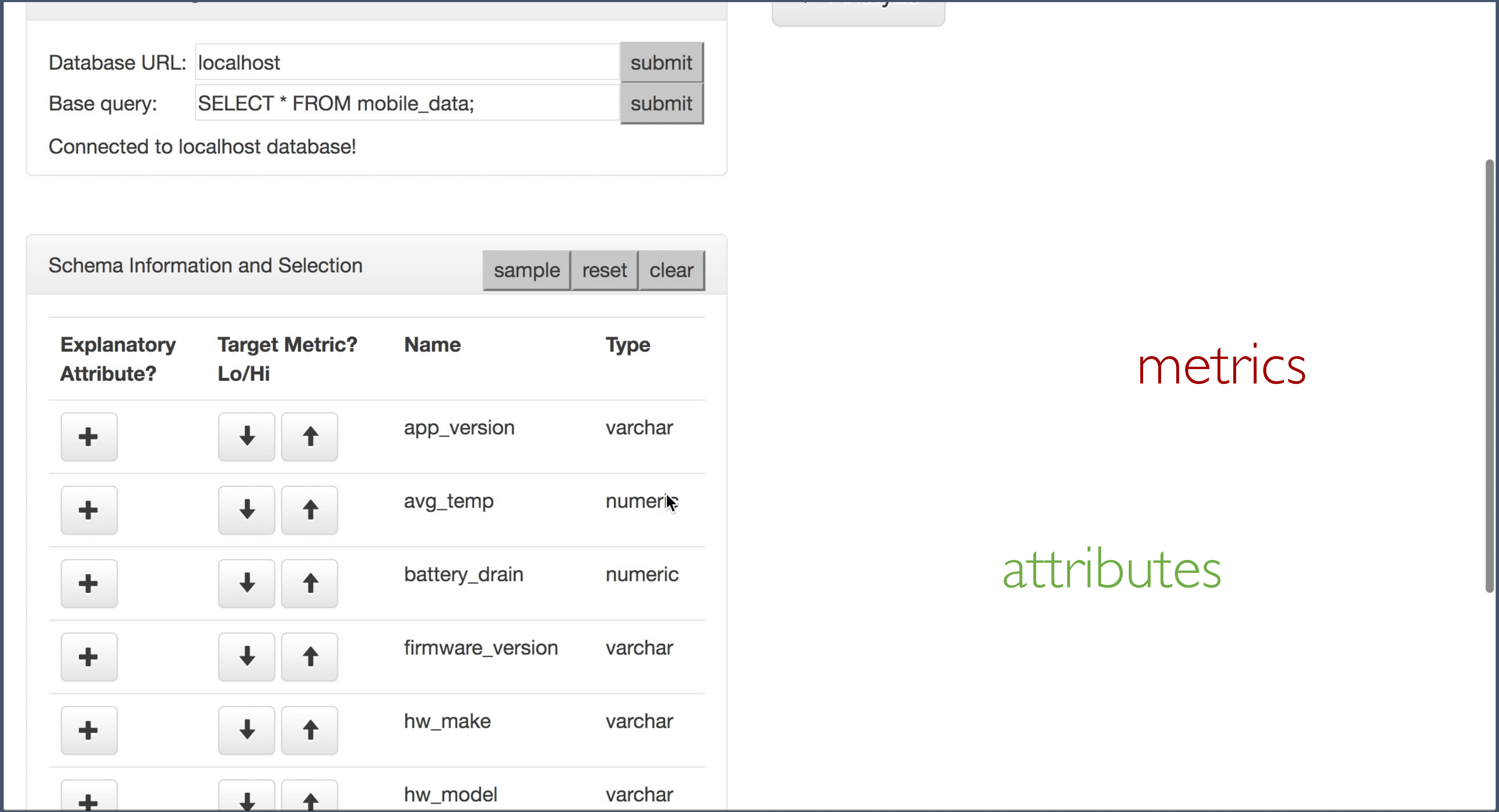
Weaknesses

- Does not provide an UI visual demonstrating a stream of explanations can be used a useful alert system.
- Did not extend the measurement noise experiment to more devices to identify statistical limit of classifier despite performance degradation.
- The description of the Real-World Dataset Accuracy involving TCP-C and TPC-E benchmarks for identifying bad OLTP servers was unclear.
- There was no breakdown of overhead b/w operators in a streaming EWS context.
- AMC may not be deployable in embedded environments.

Conclusion

Accept

MacroBase Demo



MacroBase Demo

Base query: SELECT * FROM mobile_data; submit

Connected to localhost database!

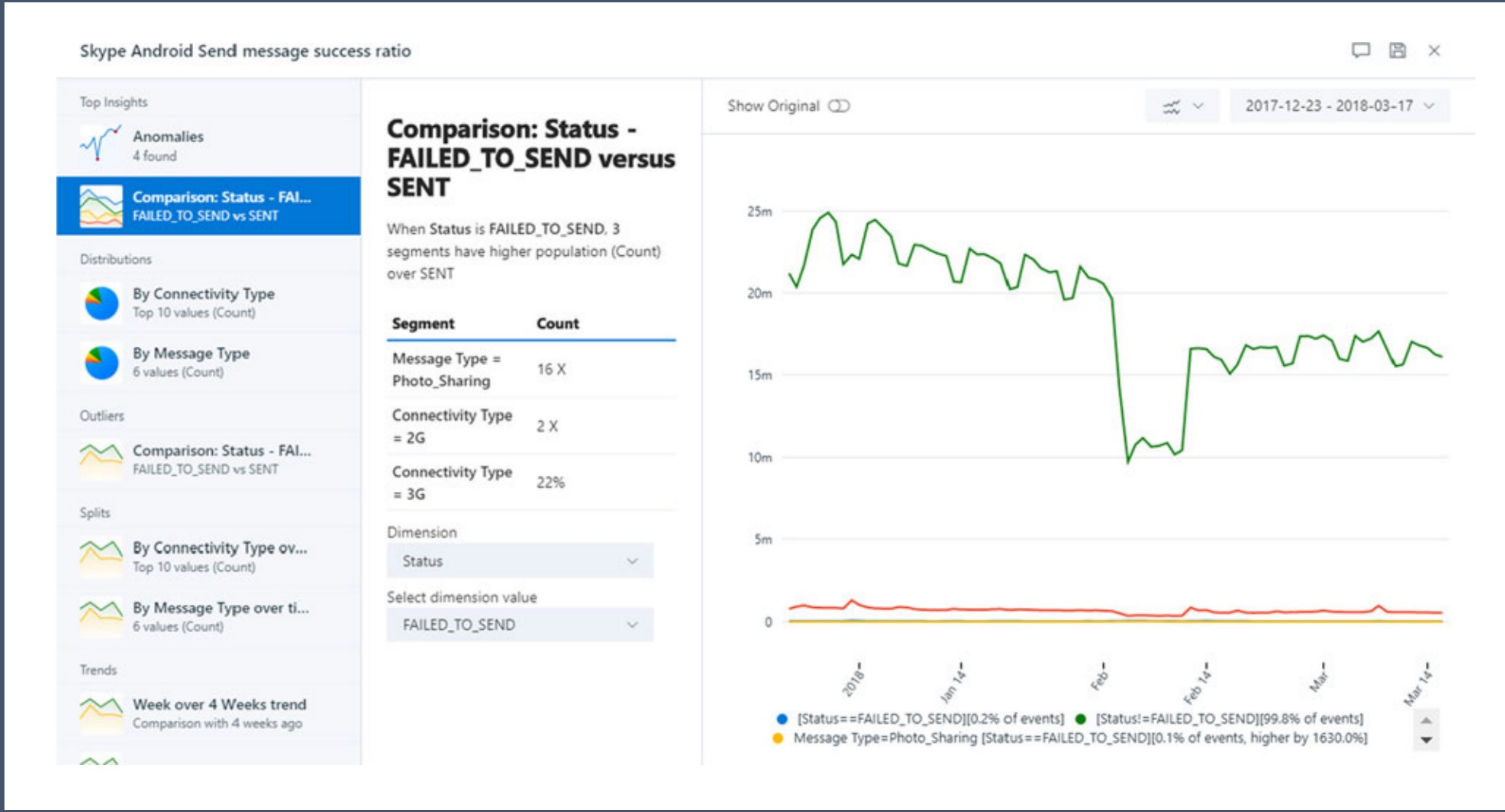
Schema Information and Selection — sample | reset | clear

Explanatory Attribute?	Target Metric? Lo/Hi	Name	Type
✓	↓ ↑	app_version	varchar
+	↓ ↑	avg_temp	numeric
+	↓ ↑	battery_drain	numeric
+	↓ ↑	firmware_version	varchar
✓	↓ ↑	hw_make	varchar
✓	↓ ↑	hw_model	varchar
+	↓ ↑	record_id	int8

DIFF: Relational Interface

```
SELECT * FROM
(SELECT * FROM logs WHERE crash = true) crash_logs
DIFF
(SELECT * FROM logs WHERE crash = false) success_logs
ON app_version, device_type, os
COMPARE BY risk_ratio >= 2.0, support >= 0.05
MAX ORDER 3;
```

How's MacroBase used?



MacroBase’s backstory

The SIGMOD submission was rated ”best of conference”

But the previous submission was rejected with low ratings

What changed?

Next class

Slice Finder: Automated Data Slicing for Model Validation

Authors: Andrew

Archaeologist: Qiandong

Practitioner: Bojun