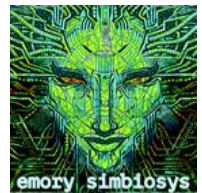
CENSUS: Counting Interleaved Workloads on Shared Storage

36th International Conference on Massive Storage Systems and Technology (MSST 2020)

Si Chen, Jianqiao Liu, Avani Wildani

EMORY UNIVERSITY Computer Science and Informatics



How to choose the right storage for workload?

Cost efficiency: higher throughput, less latency, less cost

Sequential write → LSM-tree based Key-value store

Fast random read → Flash memory

Random write → SSD

Lower speed read and write → HDD

…

And the best configuration?

Fair Resource Provisioning for Shared Storage is hard!

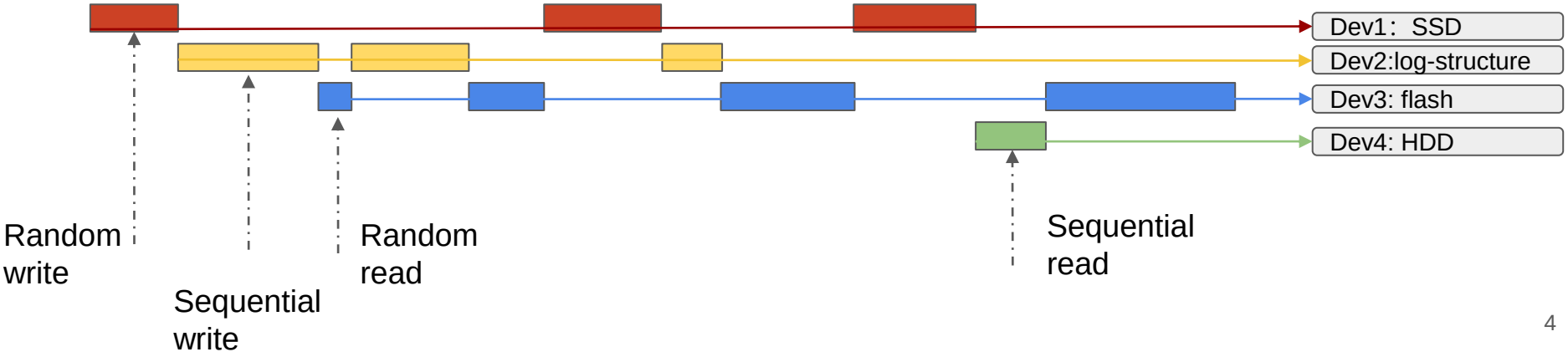
Challenge: shared storage, dynamic, interleaved,

Smart storage: capacity prediction and performance management

Deep understanding the workload!

Workload separation for shared storage

Workload
Separation !



What exactly shall we separate?

Application specific workload

Fully isolation does not really means shared storage.

Single workload has several functional usage of storage.

Functionally distinct usage of a storage system

Fworkload

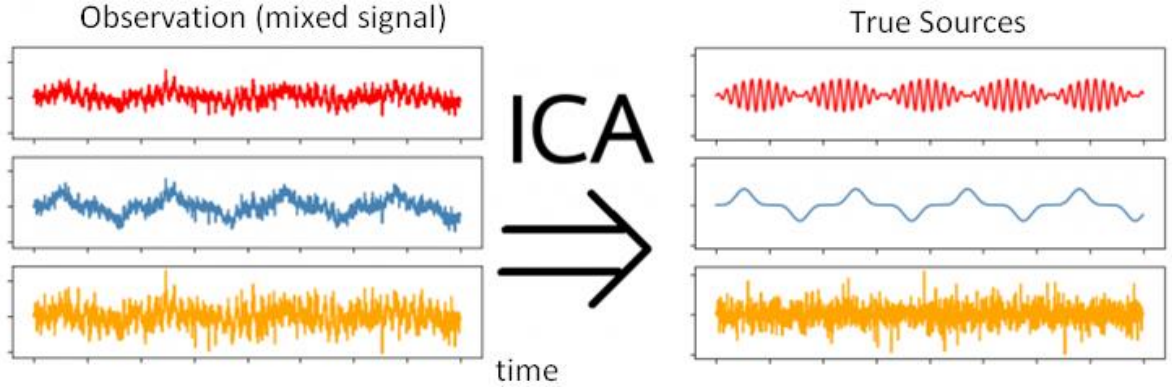
Process ID (PID) is a stand-in for non-existent labels

Motivation

Existing approaches fail to distinguish interleaved storage fworkloads.

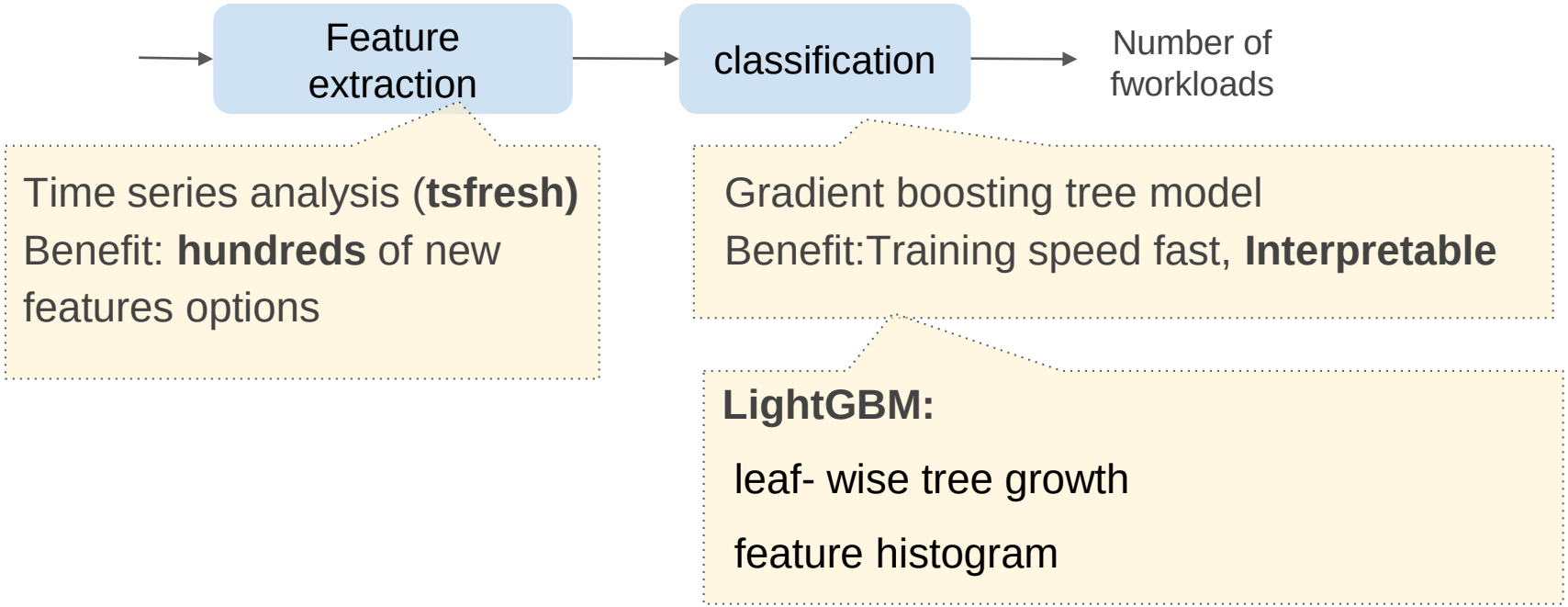
Traditional workload characterization only have limited features.
(read/write ratio, sequentiality...)

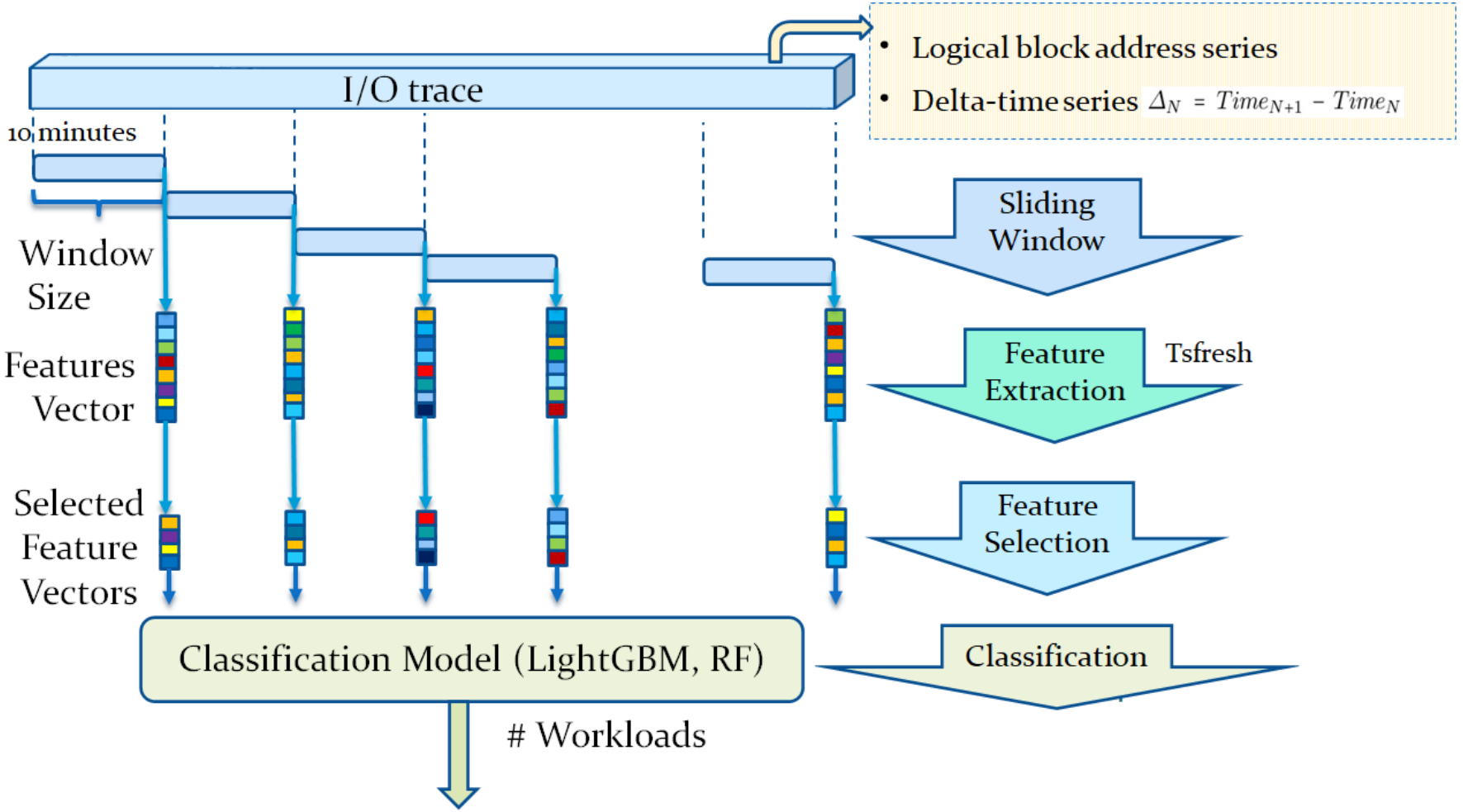
The **number** of concurrent fworkloads is precursor for separation

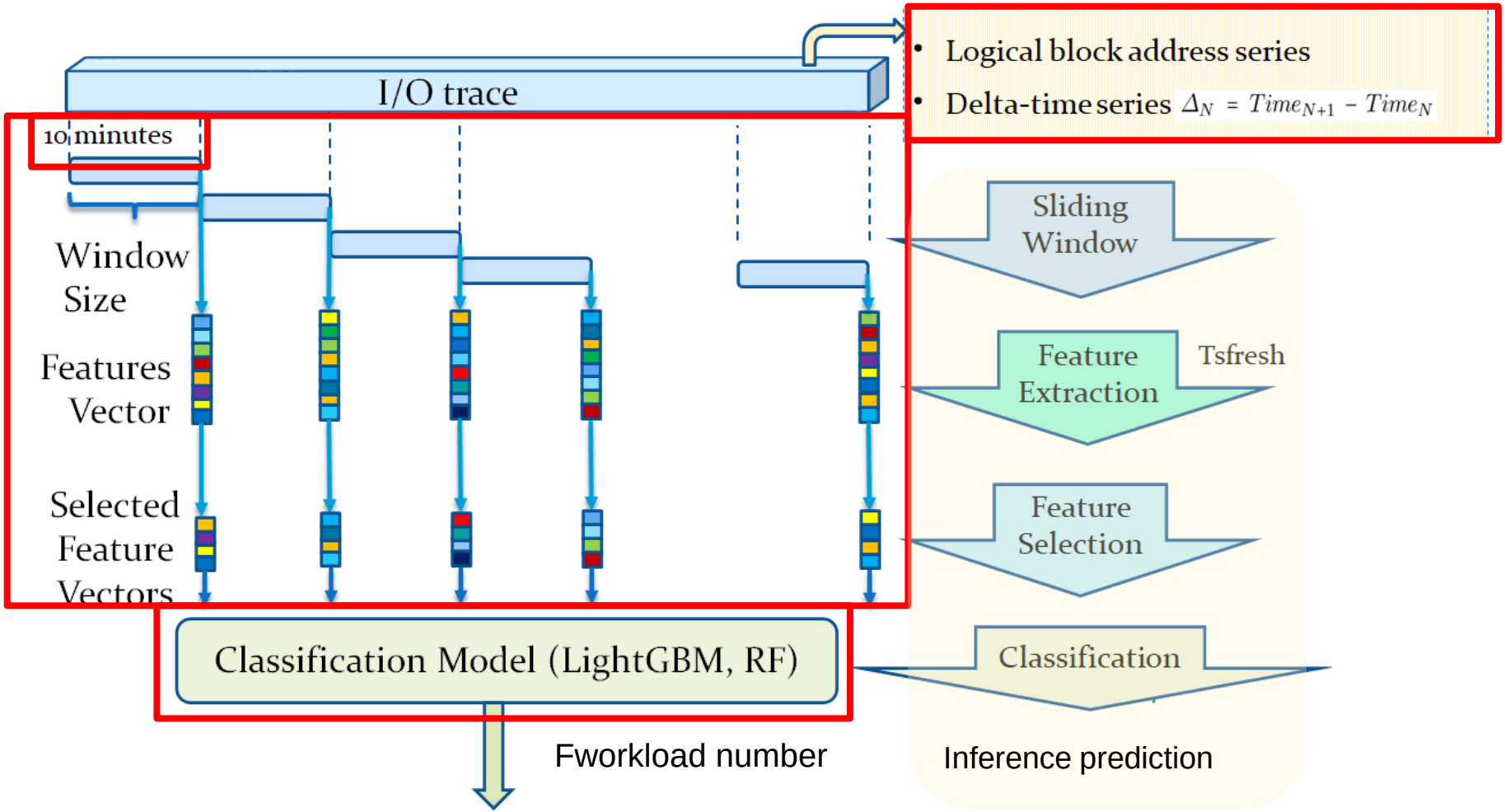


Goal: Given a block I/O trace, we are able to identify the **number** of fworkloads in a storage system.

Our Approach: Census




I/O trace
Logical block address series
Delta-time series $\Delta_N = Time_{N+1} - Time_N$
10 minutes
Window Size
Features Vector
Selected Feature Vectors
Sliding Window
Feature Extraction
Tsfresh
Feature Selection
Classification Model (LightGBM, RF)
Classification
Workloads


I/O trace
Logical block address series
Delta-time series $\Delta_N = Time_{N+1} - Time_N$
10 minutes
Window Size
Features Vector
Selected Feature Vectors
Sliding Window
Feature Extraction
Tsfresh
Feature Selection
Classification
Classification Model (LightGBM, RF)
Fworkload number
Inference prediction

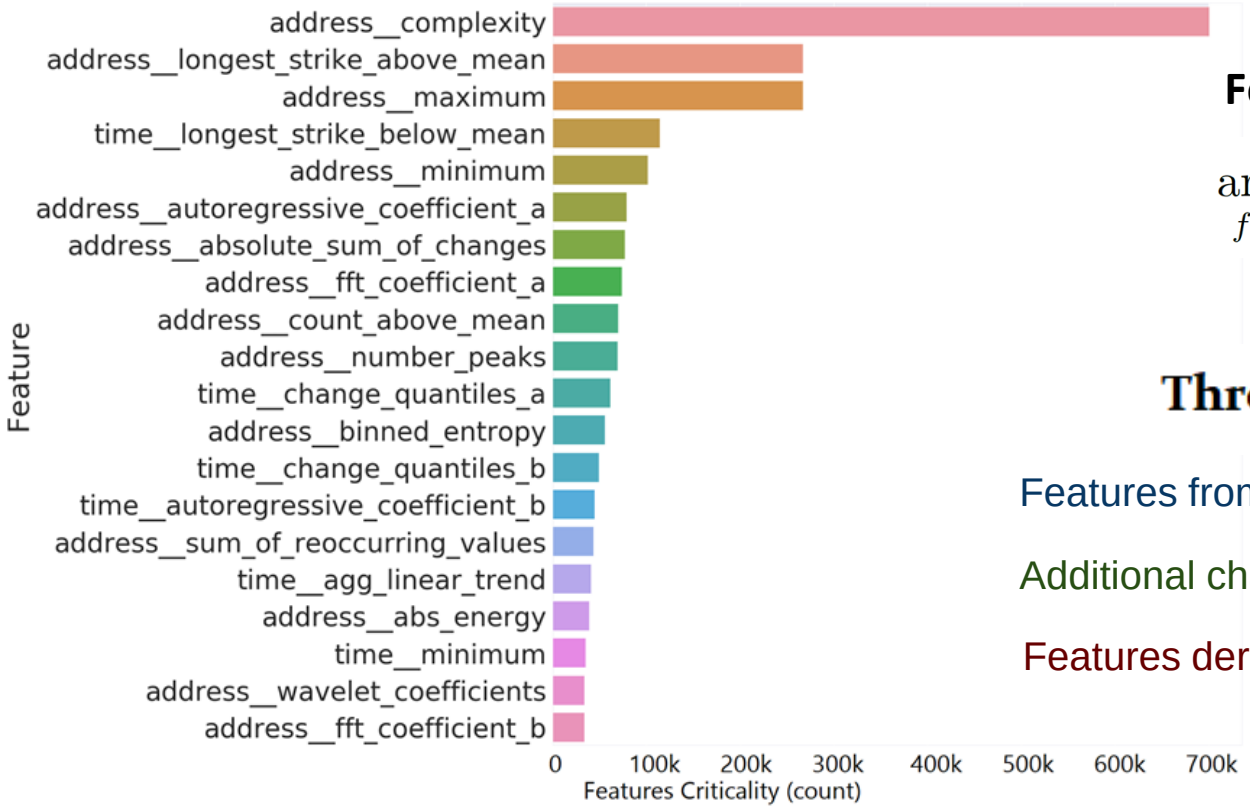
Dataset

- **FIU** (Florida International University)
  nearly three weeks of block I/O traces. Include web related, home related domain.

- **MSR** (Microsoft Research (MSR), Cambridge)
  1 week of block I/O traces from 36 different volumes on 13 enterprise servers

- **EmoryML (**newly collected**)**
  30 days of block I/O traces collected by blktrace from our local server, running machine learning workloads

Extracted features



Feature criticality = the count of

$$\arg\max_{feature} \sum_{Tree} Gain(Tree, feature)$$

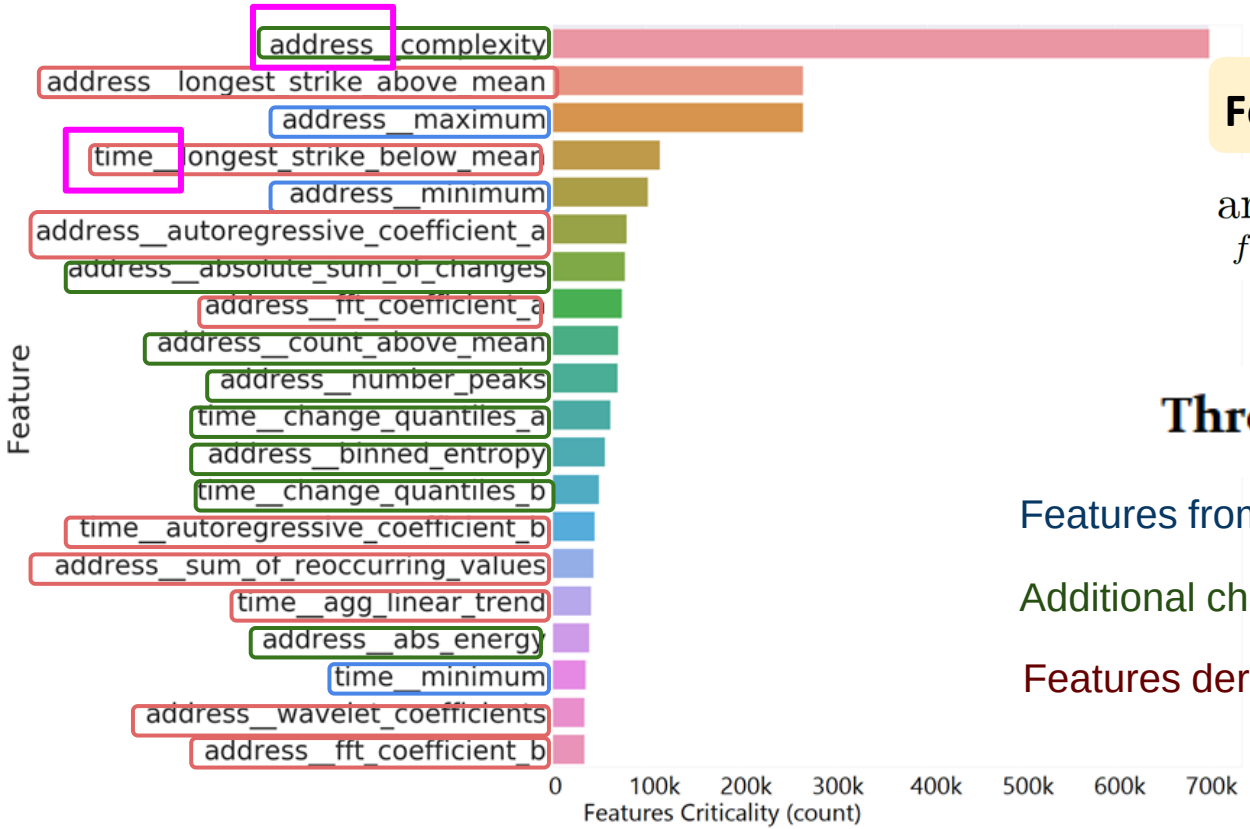
Three kinds of features

Features from summary statistics

Additional characteristics of sample distribution

Features derived from observed dynamics

Extracted features



Feature criticality = the count of

$$\arg\max_{feature} \sum_{Tree} Gain(Tree, feature)$$

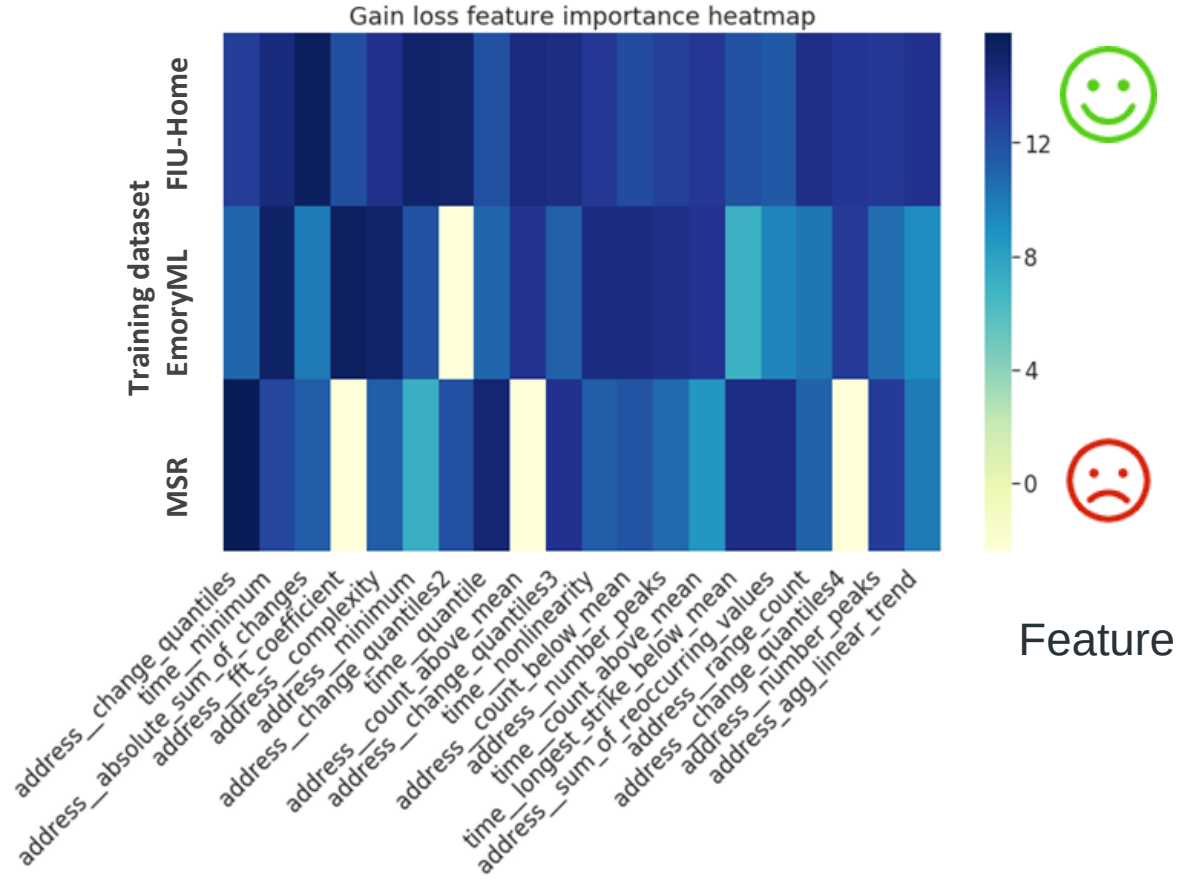
Three kinds of features

Features from summary statistics

Additional characteristics of sample distribution

Features derived from observed dynamics

Feature Importance Heatmap



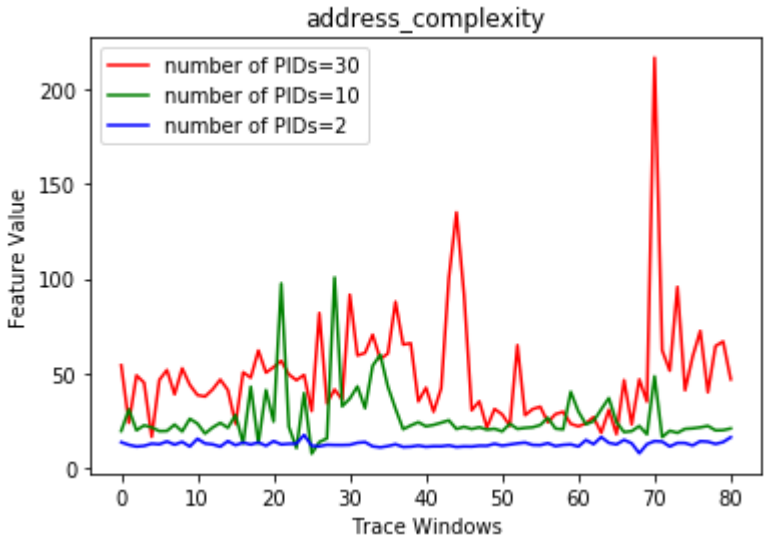
Feature criticality is trace dependent.

Sample features 1) address complexity

It measures the complexity of the address series

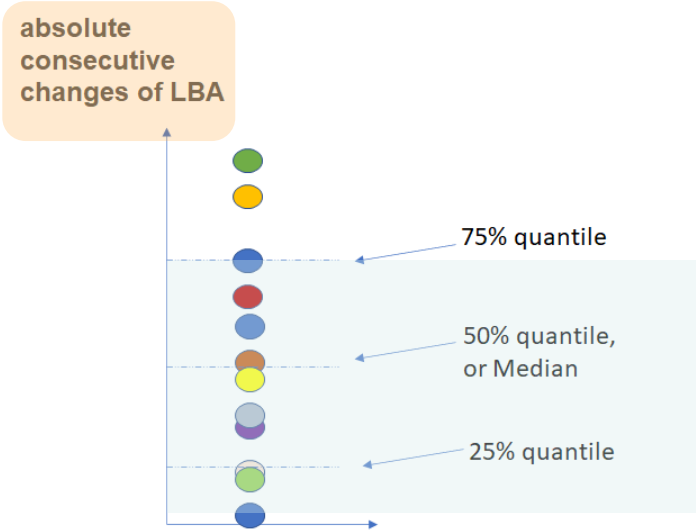
$$\sqrt{\sum_{i=0}^{n-2} (x_i - x_{i+1})^2}$$

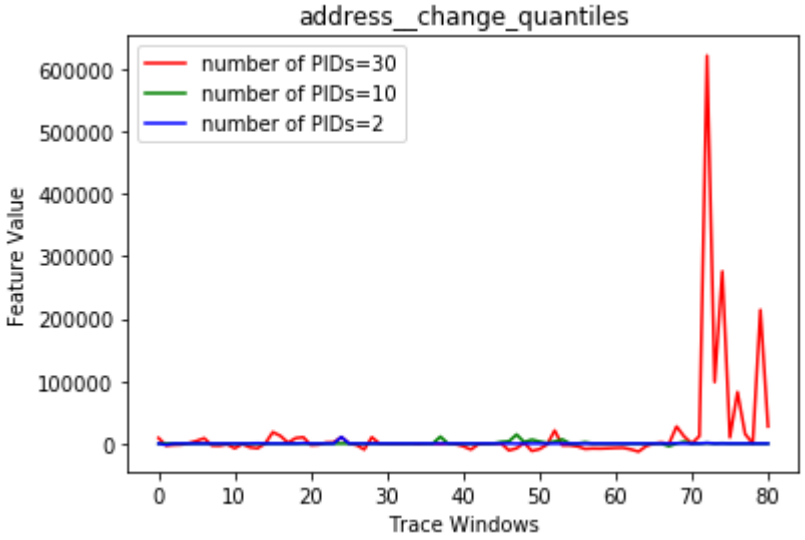
A **high** feature value indicates that **more random accesses** and less sequential accesses are in the trace, which implies **more** concurrent workloads during that time window.



Sample features 2) address change quantiles

Quantiles: divide data into equally sized groups.





It returns the **average absolute consecutive changes** of the address series identified between given higher and lower **quantiles**.

Model Evaluation

x-accuracy

Considers the instances with prediction error within 1 or 2, respectively as accurate.

MAPE (mean absolute percentage error)

Measures the size of the prediction error.

Identifies instances that are approximately correct.

$$\mathrm{M} = 100 \times \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

Baseline (fairest guess):

Randomly generating labels based on the fworkload number distribution in the training set.

Training method

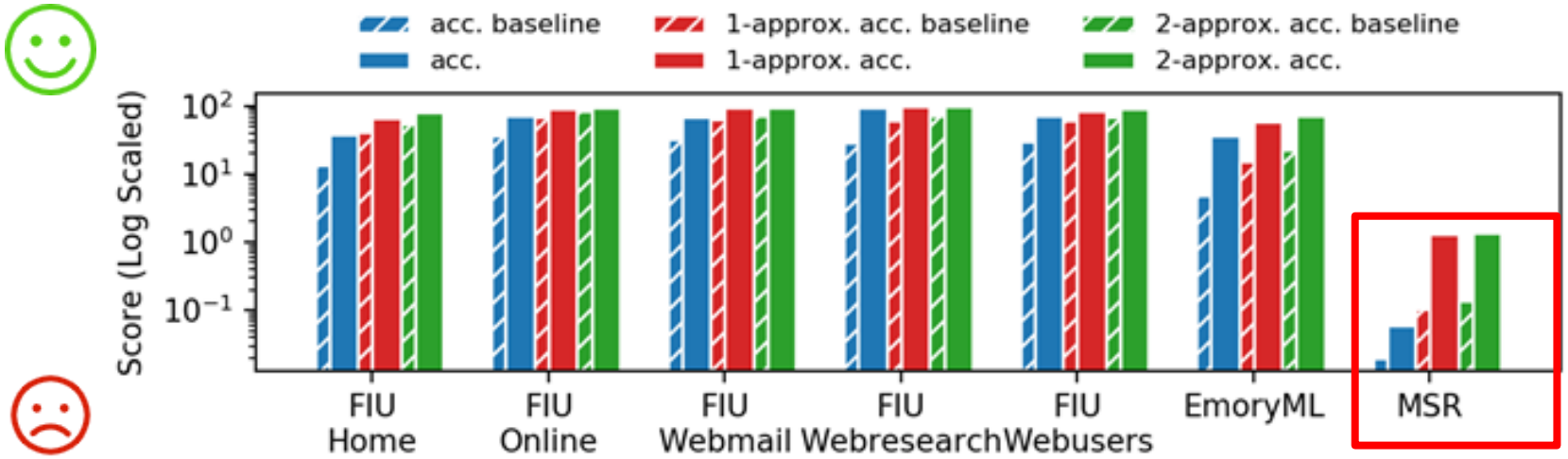
Generalized model:

Consider multiple domains

ID model:

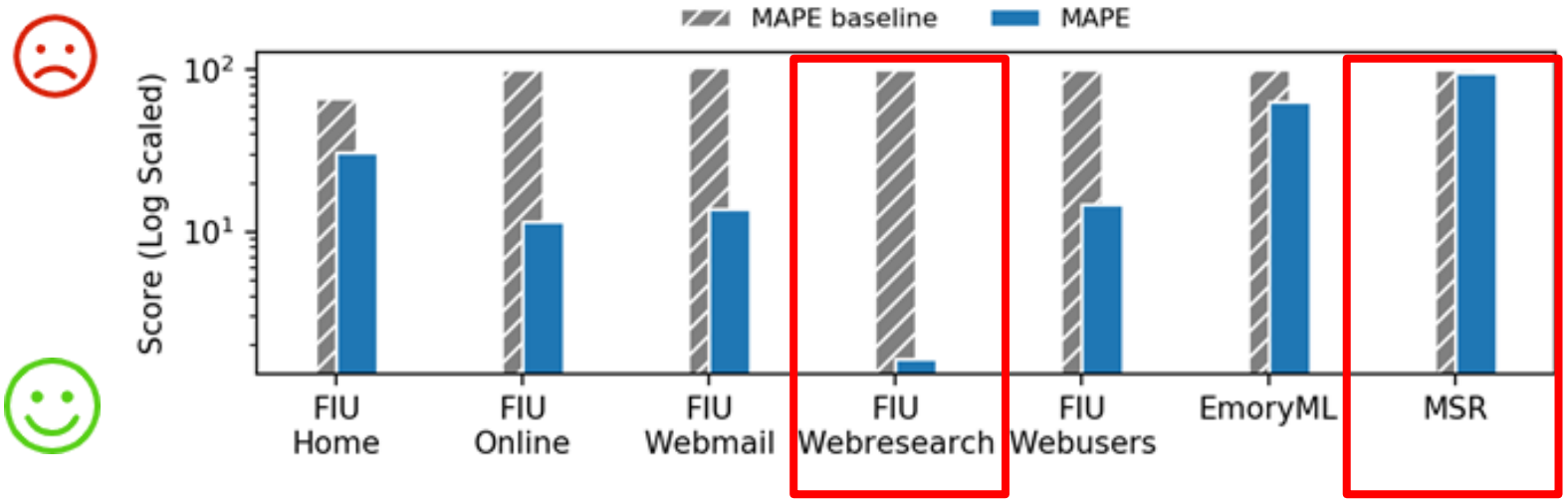
Domain specific

Result of Generalized model



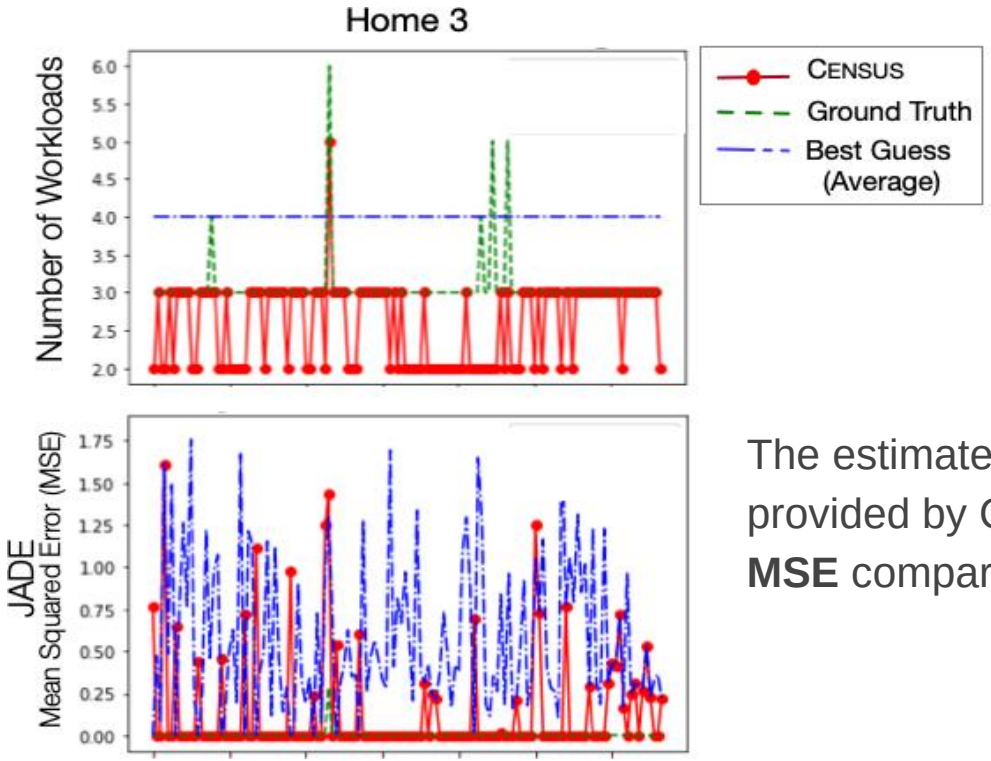
Accuracy score: CENSUS is 23% higher than baseline on average

Result of Generalized model



MAPE: CENSUS is 57% better than baseline on average

Application: Separating Interleaved fworkloads



The estimate for the number of fworkloads provided by CENSUS **decreases the average MSE** compared to the fair guess MSE

Summary

- CENSUS could **identify the number** of concurrent fworkloads with as little as 5% error.
- CENSUS opens the field to insights derivable from formerly overlooked **metrics**.
- LBA carries more effective information than **time interval**. Only 30% top features are related to time, affecting 1% of the final result.
- CENSUS improves fworkload separation in a test case.

Discussion and Future work

- Online model, recurrently training the model when unknown fworkload emerge.
- Find better fworkload label instead of PID, e.g. UID, process name.
- Add more trace attributes for workload characterization, e.g. latency.
- Try the workload separation on large-scale dataset.

Thank you! Questions!

si.chen2@emory.edu
https://github.com/meditates/CENSUS