Fast Outlier Detection in Oblique Subspace

Bowen Li

Oct. 22, 2025

Publication & Authorship

Bowen Li

Seminars of Scientific Computation

Publication

- **Conference**: ACM CIKM 2025 (Nov. 10–14, 2025, Seoul, Korea)
- **Paper Title**: Fast Outlier Detection in Oblique Subspaces
- **Authors**: Bowen Li, Charu C. Aggarwal, Peixiang Zhao
- **Affiliations**: Florida State University, IBM T. J. Watson Research Center

CIKM SEOUL 2025

November 10 - 14

Authors

- **Name**: Bowen Li
  - A final-year PhD student, Computer Science
- **Advisor**: Prof. Peixiang Zhao
- **Research Interests**: Data mining, large-scale database systems, and learning–driven solutions to fundamental data problems.

Co-Authors

- **Dr. Peixiang Zhao**
- Full professor
- Computer Science, Florida State University

- **Dr. Charu C. Aggarwal**
- Distinguished Research Staff Member
- IBM T. J. Watson Research Center

Introduction

Bowen Li

Seminars of Scientific Computation

Problem Statement

- **Problem**: outlier detection
- **Definition**:

In high-dimensional datasets, an **outlier** is a data object deviating significantly from the general patterns of underlying data, often appearing distant or unusual to other objects (a.k.a. **inliers**).

- **Reasons**:
  - Natural variation in the data
  - Mistakes or noise during data collection
  - Rare or unusual events that carry important insights

Applications

- Intrusion identification
- Medical diagnosis
- Financial fraud detection
- Traffic management
- And so on …

Challenge

- Curse of dimensionality
- High computational cost (e.g., $O(n^2)$ or more).
- Dependence on predefined attributes or vector representations.

Related Work

Bowen Li

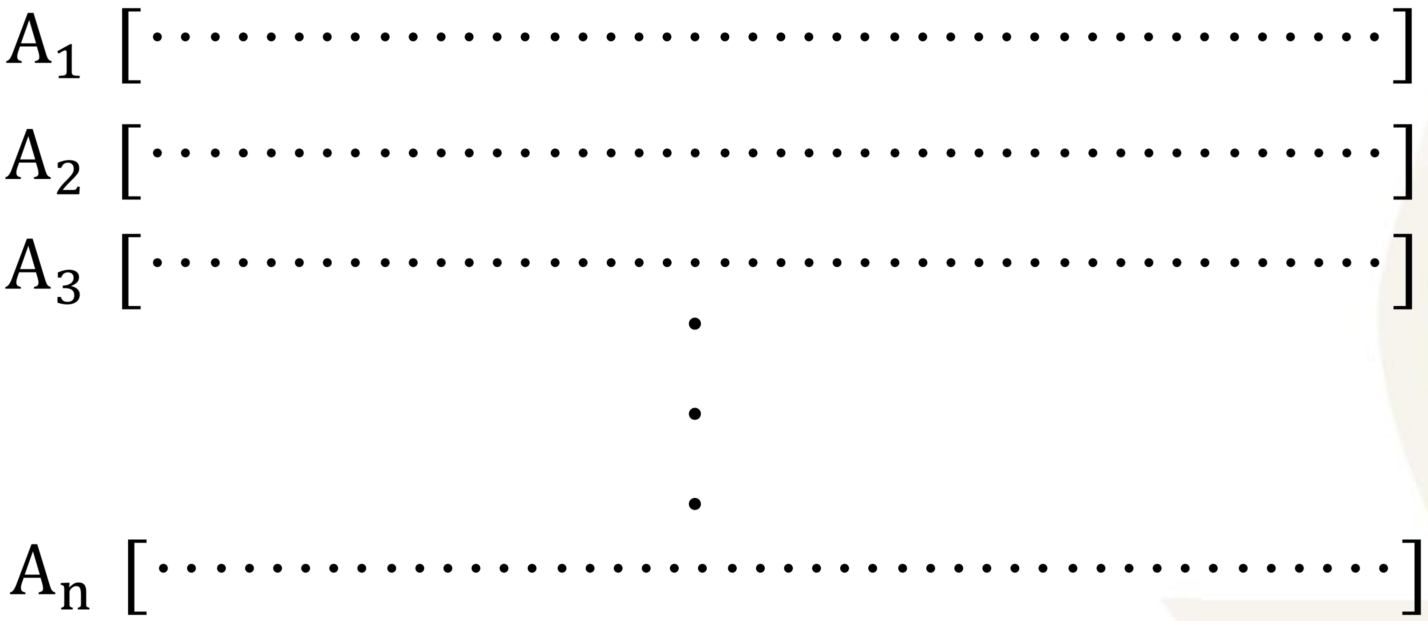
Seminars of Scientific Computation

Categories

- Statistical methods
- Distance-based methods
- Density-based methods
- Pattern compression methods
- Spectral methods
- Subspace methods

RS-Hash

- A subspace hashing method



A_1

A_2

A_3

A_n

Dataset *D*

RS-Hash: Sampling

- Sampling **m** (m ≤ n)points randomly

S_1 $[\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots]$

S_2 $[\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots]$

S_3 $[\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots]$

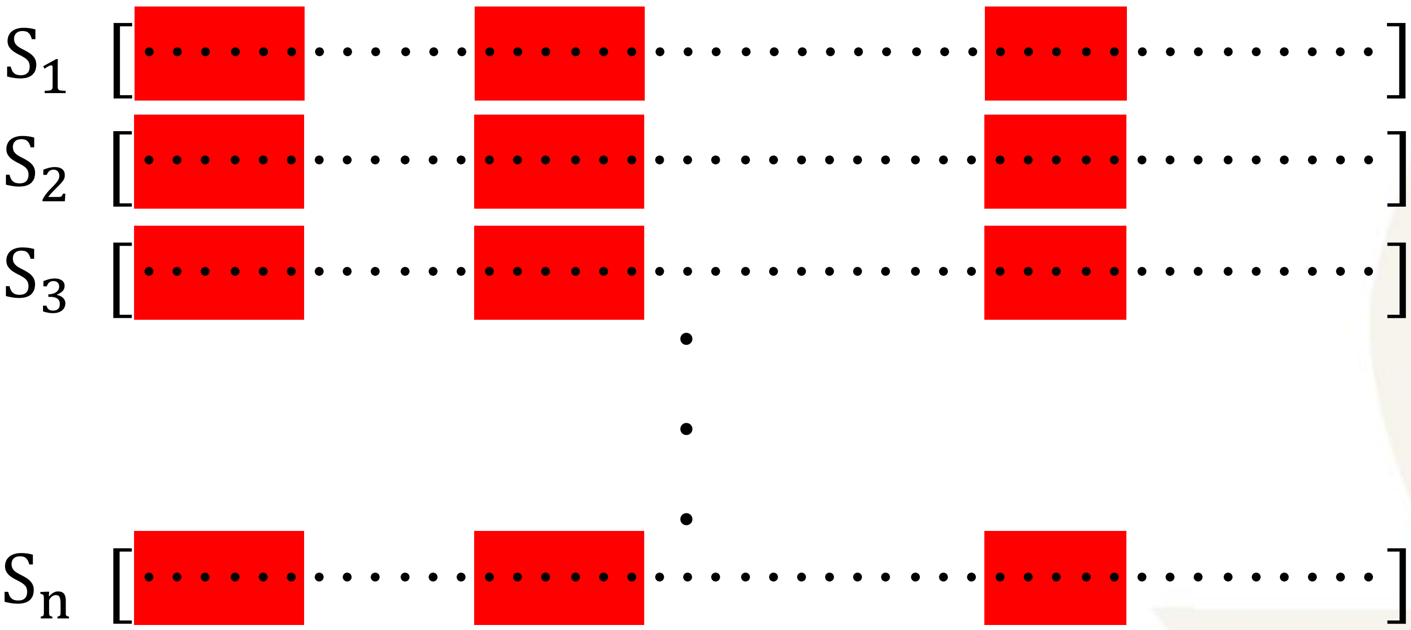
$\vdots$

S_m $[\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots]$

Sample *S*

RS-Hash: Subspace

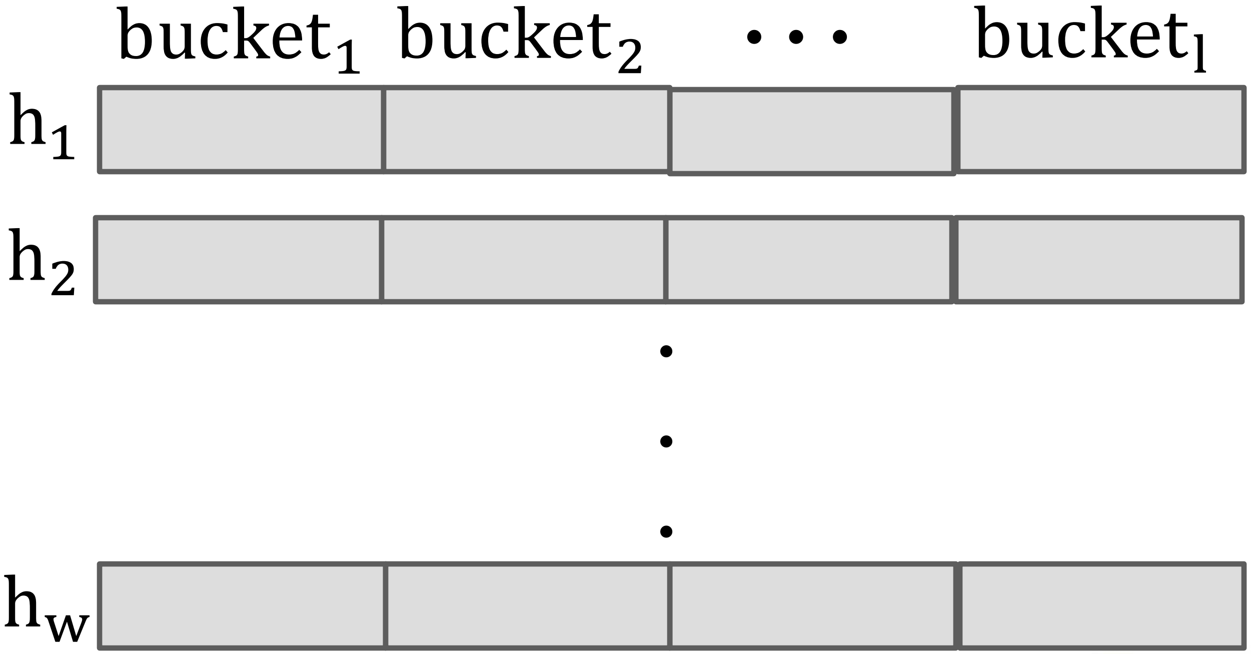
- Select **r** subspaces randomly



Subspace Histogram *R*

RS-Hash: Hashing

- Train a **Count-Min Sketch**



bucket_1 bucket_2 $\cdots$ bucket_l

h_1

h_2

$\vdots$

h_w

Count-Min Sketch *H*

Limitation

- Designed for multidimensional data with *pre-defined dimensions or attributes*
- Limited to *axis-parallel subspaces*

- How about those *arbitrary-shaped or schema-less data* without explicit dimensions?

Oblique Subspaces

Bowen Li

Seminars of Scientific Computation

Problem Setting

- A data collection $\boldsymbol{O}=\{\mathrm{O}_1, \mathrm{O}_2, \dots, \mathrm{O}_\mathrm{n}\}$
  - $\mathbf{n}$ objects: multidimensional vectors, graphs, time series and, so on.
  - Similarity function: $\mathbf{S_{ij}=s(O_i, O_j)}$
    - Multidimensional vectors: L2 distance-based similarity
    - Time series: dynamic time wrapping (DTW)
    - Graphs: graph-kernel based similarity

Oblique Vector Direction

- Consider a pair of objects (O_i, O_j) from O to construct an ***oblique*** vector direction
- For the rest of the objects $\mathbf{O_k}$, the projection on the oblique vector direction can be defined as:

$$\begin{aligned}\mathbf{proj}(O_k) &= (\overrightarrow{X_k} - \overrightarrow{X_i}) \cdot (\overrightarrow{X_j} - \overrightarrow{X_i}) \\ &= \overrightarrow{X_k} \cdot \overrightarrow{X_j} - \overrightarrow{X_k} \cdot \overrightarrow{X_i} - \overrightarrow{X_i} \cdot \overrightarrow{X_j} + \overrightarrow{X_i} \cdot \overrightarrow{X_i} \quad \mathbf{(1)} \\ &= s_{kj} - s_{ki} - s_{ij} + s_{ii}.\end{aligned}$$

- picking r pairs of objects → an r-dimensional oblique subspace of O → create a histogram in this oblique subspace

OS-Hash

Bowen Li

Seminars of Scientific Computation

OS-Hash

1. Parameter Selection
2. Oblique Subspace Identification
3. Sample Data Projection
4. Oblique Subspace Hashing
5. Outlier Score Evaluation

Step 1: Parameter Selection

- **Sampling size s**
  - a small constant
  - $s = min\{ n, 1000\}$
- **Locality parameter f**
  - defines the bucket width as a fraction of the length along each oblique dimension
  - $f = (1/\sqrt{s}, 1 - 1/\sqrt{s})$
- **Oblique subspace dimensionality r**
  - $[1 + 0.5\lfloor log_{max\{2,1/f\}}(s)\rfloor, \lceil log_{max\{2,1/f\}}(s)\rceil]$

Step2: Oblique Subspace Identification

- Sample $\mathbf{r}$ pairs of objects at random from $\boldsymbol{O}$ randomly
  - $\mathbf{i}$-th pair of objects, $(\boldsymbol{O}_{\mathrm{a_i}}, \boldsymbol{O}_{\mathrm{b_i}})$
  - $\mathbf{i}$-th oblique dimension

- $\mathbf{r}$-dimensional oblique subspaces for $\boldsymbol{O}$

Step 3: Sample Data Projection

- Select a sample $\boldsymbol{S} \subseteq \boldsymbol{O}$ of objects at random, where $|\mathrm{S}| = s$
- Project the objects of $\mathbf{S}$ along each of r oblique dimensions specified by $(\mathbf{O}_{\mathbf{a_i}}, \ \mathbf{O}_{\mathbf{b_i}})$ according to Equation 1.

- Each sample object $\mathbf{O_i} \in \mathbf{S}$ is represented as an r-dimensional vector in the oblique subspace
  - j-th oblique dimension: $\mathbf{z_{ij}}$

Step 3: Cont'd

- normalize $\mathbf{z_{ij}}$ to $\mathbf{z'_{ij}}$

$$z'_{ij} \Leftarrow \frac{z_{ij} - \min_j}{\max_j - \min_j} \quad \textbf{(2)}$$

- a normalized r-dimensional vector of sampled object $\mathbf{O_i}$

$$\overrightarrow{Z'_i} = (z'_{i1}, \ldots, z'_{ir})$$

Step 3: Cont'd

- Create for each $\mathbf{O_i}$ a new r-dimensional discrete vector $\overrightarrow{\mathbf{Y_i}}$
  - $y_{ij} = \lfloor (z'_{ij} + \alpha_j)/f \rfloor$
    - $\boldsymbol{\alpha}_\mathbf{j}$ is a shift parameter drawn uniformly at random from $(\mathbf{0},f)$
    - address the edge effects in the first and last buckets of the histogram
  - the integer bucket values of $\mathbf{O_i}$ using the fractional width f for each bucket
- Define a histogram representing all the r oblique dimensions.

Step 4: Oblique Subspace Hashing

- Construct a Count-Min sketch $\mathbf{H}$
  - Width $\mathbf{w}$: w hash tables implementing w pairwise independent hash functions
  - Length $\mathbf{l}$: the number of elements of these hash tables
  - **Input**: r-dimensional bucket vector $\overrightarrow{\mathbf{Y_i}}$
  - **Output**: integer value in the range of $(\mathbf{0}, \boldsymbol{l}\mathbf{-1})$

- Apply each hash function $\mathbf{H_k}$ upon $\overrightarrow{\mathbf{Y_i}}$,
  - increment the count value in $\mathbf{H_k}(\overrightarrow{\mathbf{Y_i}}) - \mathrm{th}$ bucket by 1

Step 5: Outlier Score Evaluation

- Transform each object $\mathbf{O_i} \in \mathbf{O}$ into its r-dimensional bucket representation $\overrightarrow{\mathbf{Y_i}}$
  - Approximation: The values of $\mathbf{min_j}$ and $\mathbf{max_j}$ are derived from the sample $\mathbf{S}$
- Insert $\overrightarrow{\mathbf{Y_i}}$ into the constructed Count-Min Sketch $\mathbf{H}$
  - $\mathbf{c_k}$: the value of the $\mathbf{H_k}(\overrightarrow{\mathbf{Y_i}})$ -th cell in the k-th hash table
    - $\mathbf{O_i} \in \mathbf{S}$: $score(O_i) = log_2(min\{c_1, \ldots, c_w\})$
    - $\mathbf{O_i} \notin \mathbf{S}$: $score(O_i) = log_2(min\{c_1, \ldots, c_w\} + 1)$

Step 5: Cont'd

- A single-base detector of OS-Hash is too weak
- Repeat $\boldsymbol{m}$ times, once for each base detector of the ensemble
- Let $\mathbf{os_j^i}$ represent the outlier score of the $\boldsymbol{i}$-th object from the $\boldsymbol{j}$-th base detector

$$\text{OS-Hash}(O_i) = \frac{1}{m} \sum_{j=1}^{m} os_j^i \quad \textbf{(3)}$$

Count-Min Sketch-Based Hashing

- Count-Min Sketch **H**
  - **w** hash tables: w=4
  - Hash value range **l**: l=10*s=10,000
- For each hash table:
  - success probability of a single object: $1/l$
  - No collision between **s** sampled objects: $(1-1/l)^s$
- Collison within w hash tables: $(1-(1-1/l)^s)^w$
- No collisions arise in at least one of the *w* hash tables

$$[1-(1-(1-1/l)^s)^w] \approx 0.9999$$

Complexity

- **Time Complexity**: $O(nmT \log s)$
  - **O(T):** object similarity computation
  - **O(n log s):** number of similarity computation **O(nr)**
    - O(r) → O(log s)
  - **O(m)**: number of base outlier detectors
  - Linear

- **Space Complexity**: $O(w \cdot l)$
  - Constants

OS-Hash in Data Stream

Bowen Li

Seminars of Scientific Computation

Challenge

- **Data streams**: a continuous and rapid flow of data objects arrives in real time
  - Processed in one pass
  - A large amount of data is coming quickly and continuously
  - Rapid changes of underlying patterns

- Real-time outlier detection is extremely challenging!

Techniques in Data Stream

- **Sliding Window**: Incoming objects are automatically inserted into the sketch, and obsolete ones falling off from the sliding window are removed.

- **Time-decayed model**: an exponential function with a decay rate **λ** to quantify the time-varying weight of an object $\mathbf{O_i}$
  - t objects have arrived after $\mathbf{O_i}$, the weight of $\boldsymbol{O_i}$ is $2^{-\lambda t}$

Modifications

- Each base detector is created consecutively → Maintain all ensemble components in the same Count-Min sketch:
  - The values of $\mathbf{min_j}$ and $\mathbf{max_j}$: estimated an initial sample of streaming data.
  - Sampling size $s = max\{1000, 1/\left(1 - 2^{-\lambda}\right)\}$
  - Locality parameter $\mathbf{f}$, dimensionality $\mathbf{r}$, and shift parameters $\boldsymbol{\alpha}$: calculated in the initial step at one time

Lazy Weighting Strategy

- In each Count-Min Sketch cell
  - Count
  - $\mathbf{t_l}$: last time it is updated
- When a new object is streaming in
  - $\mathbf{t_c}$: the current time stamp
  - Updated count: $c * 2^{-\lambda(t_c - t_l)} + 1$

OS-Stream

For the streaming object $\mathcal{O}$, we compute its score

1. For each base detector $i \in \{1, \cdots, m\}$, calculate the $\mathbf{r}$-dimensional bucket representation $\overrightarrow{\mathbf{Y_i}}$
2. For each hash function $\mathbf{H_k}$, compute the $\boldsymbol{H_k(\overrightarrow{Y_i})}$ to get the weighted count $\mathbf{c_k^i}$
3. The score of $\mathbf{i}$-th based detector is $log\left(1 + min\{c_1^{\mathrm{i}}, \ldots, c_w^i\}\right)$, sum them and calculate the average
4. Update both the counts and time-stamps

Experiment

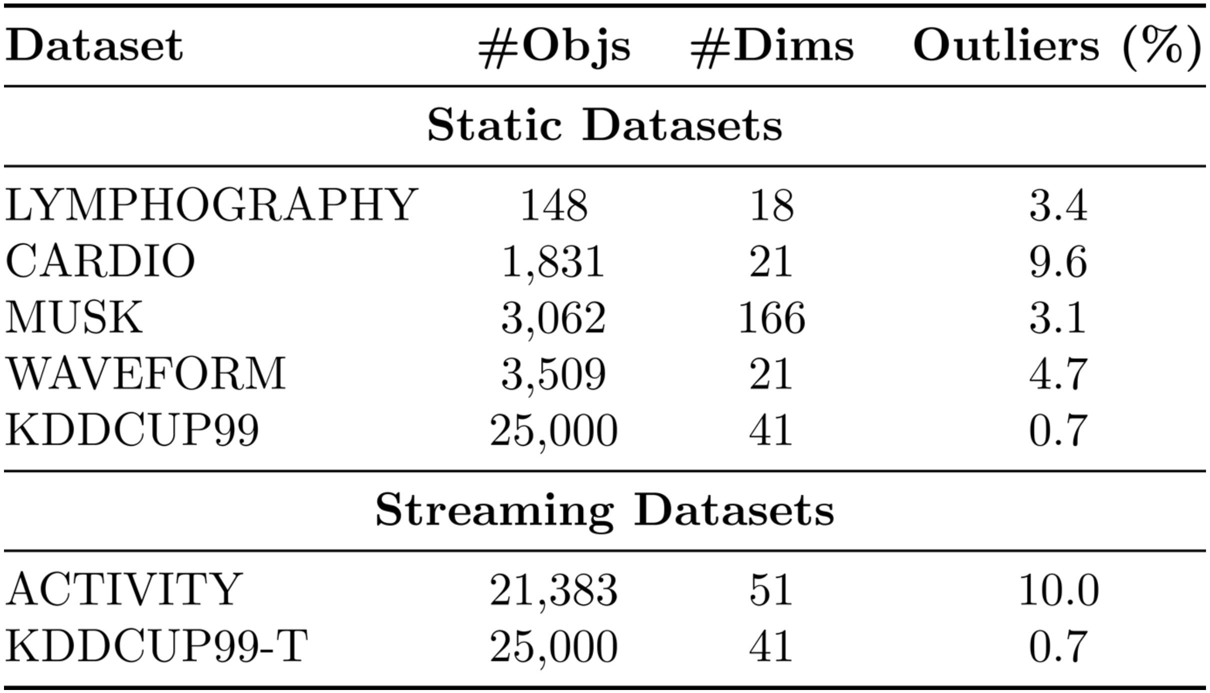
Bowen Li

Seminars of Scientific Computation

Datasets

- **Multidimensional Datasets**
  - *Static:* LYMPHOGRAPHY, CARDIO, MUSK, WAVEFORM, KDDCUP99
  - *Stream:* ACTIVITY, KDDCUP99-T,
- **Time Series Datasets**
  - *Static:* PICKUP, PEBBLE, POWER, ECG5000, CROP
  - *Stream:* ACTIVITY-T, CROP-T
- **Graph Datasets**
  - *Static:* MUTAG, FINGER, AIDS, MUTAGEN, TOX21
  - *Stream:* TOX21-AR-T, MCF-7-T

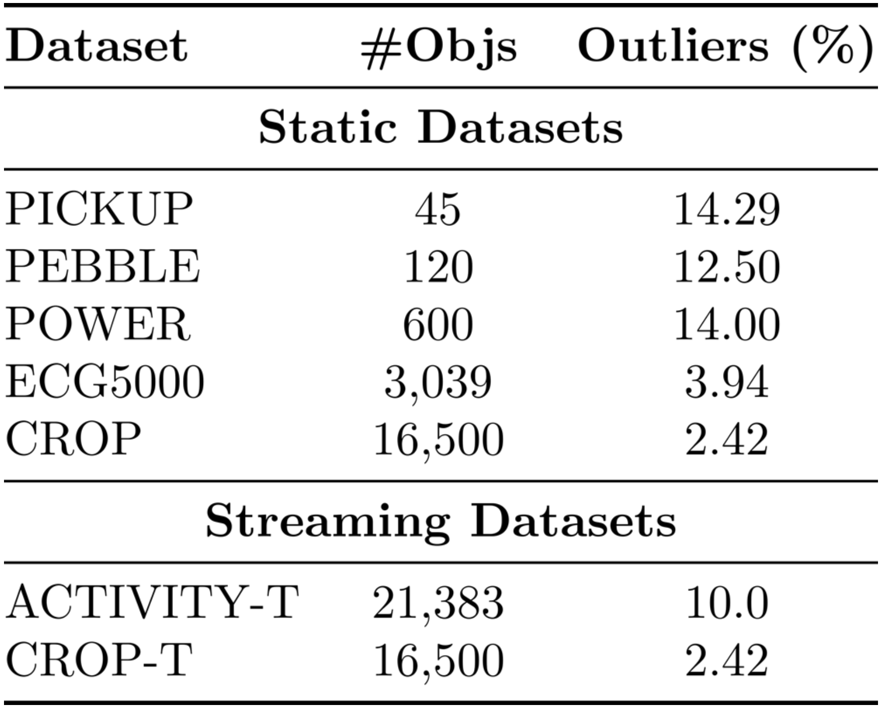
Statistics of datasets



Dataset	#Objs	#Dims	Outliers (%)
Static Datasets			
LYMPHOGRAPHY	148	18	3.4
CARDIO	1,831	21	9.6
MUSK	3,062	166	3.1
WAVEFORM	3,509	21	4.7
KDDCUP99	25,000	41	0.7
Streaming Datasets			
ACTIVITY	21,383	51	10.0
KDDCUP99-T	25,000	41	0.7

Table 1: Multidimensional Datasets

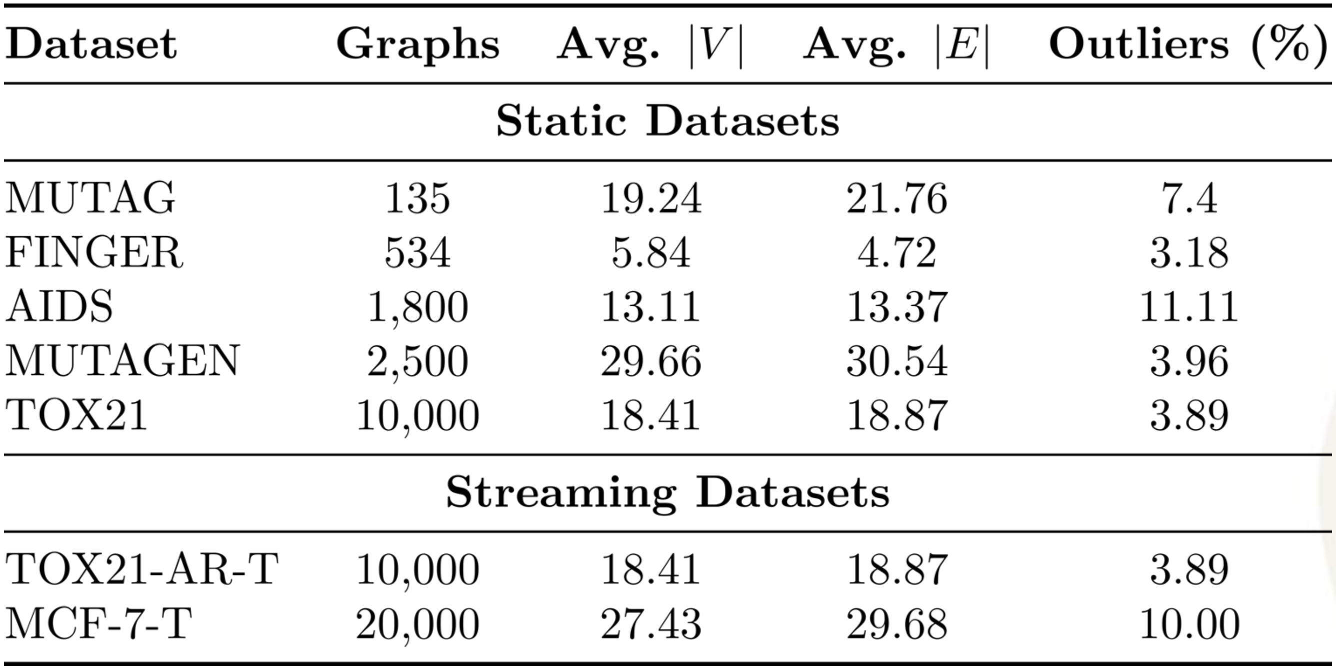
Statistics of datasets



Dataset	#Objs	Outliers (%)
Static Datasets		
PICKUP	45	14.29
PEBBLE	120	12.50
POWER	600	14.00
ECG5000	3,039	3.94
CROP	16,500	2.42
Streaming Datasets		
ACTIVITY-T	21,383	10.0
CROP-T	16,500	2.42

Table 2: Time Series Datasets

Statistics of datasets



Dataset	Graphs	Avg. $\lvert V \rvert$	Avg. $\lvert E \rvert$	Outliers (%)
Static Datasets				
MUTAG	135	19.24	21.76	7.4
FINGER	534	5.84	4.72	3.18
AIDS	1,800	13.11	13.37	11.11
MUTAGEN	2,500	29.66	30.54	3.96
TOX21	10,000	18.41	18.87	3.89
Streaming Datasets				
TOX21-AR-T	10,000	18.41	18.87	3.89
MCF-7-T	20,000	27.43	29.68	10.00

Table 3: Graph Datasets

Evaluation Metrics

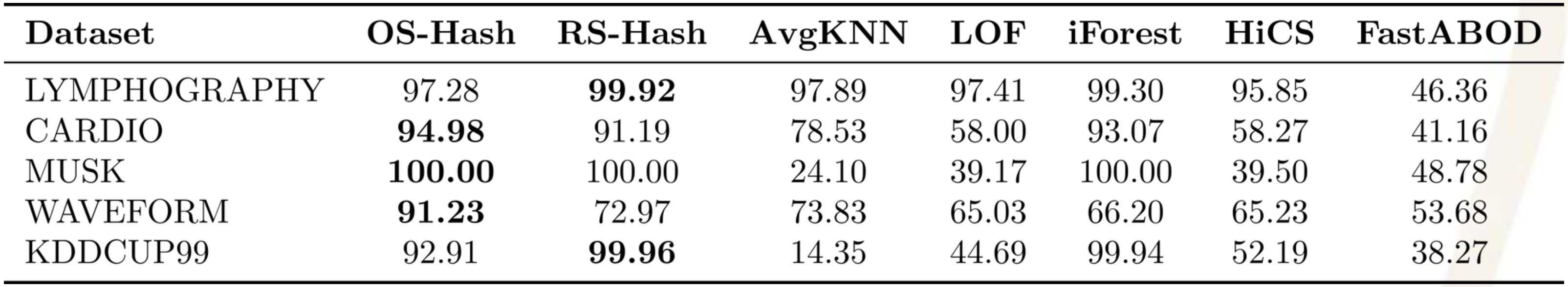
- **Effectiveness**
  - Area Under the Curve (AUC) Score
  - *Static*: Receiver Operating Characteristics (ROC) Curve
- **Efficiency**
  - *Static:* overall runtime (in seconds)
  - *Stream:* the number of objects processed per second

Baselines

- **Multidimensional Datasets**
  - *Static:* AvgKNN, FastABOD, iForest, HiCS, LOF, and RS-Hash
  - *Stream:* RS-Stream, LOF-Stream, and AvgKNN-Stream

- **Time Series and Graph Datasets**
  - *Static:* AvgKNN, LOF, COF, LoOP
  - *Stream:* LOF-Stream and AvgKNN-Stream

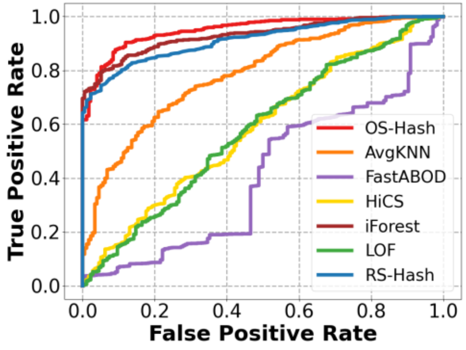
Static: Multidimensional Datasets



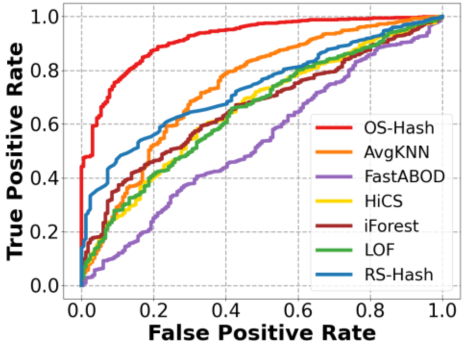
Dataset	OS-Hash	RS-Hash	AvgKNN	LOF	iForest	HiCS	FastABOD
LYMPHOGRAPHY	97.28	**99.92**	97.89	97.41	99.30	95.85	46.36
CARDIO	**94.98**	91.19	78.53	58.00	93.07	58.27	41.16
MUSK	**100.00**	100.00	24.10	39.17	100.00	39.50	48.78
WAVEFORM	**91.23**	72.97	73.83	65.03	66.20	65.23	53.68
KDDCUP99	92.91	**99.96**	14.35	44.69	99.94	52.19	38.27

Table 4: AUC results for multidimensional datasets

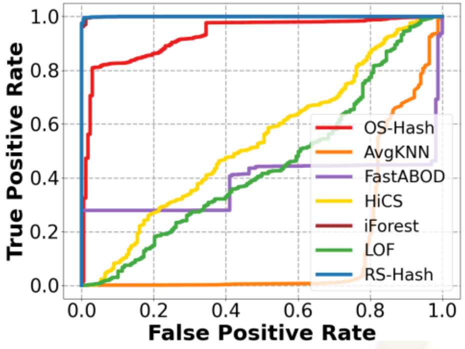
Static: Multidimensional Datasets



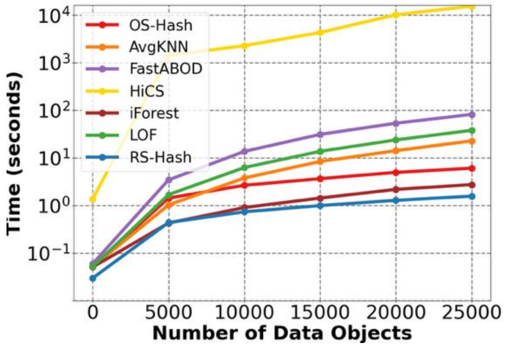
CARDIO



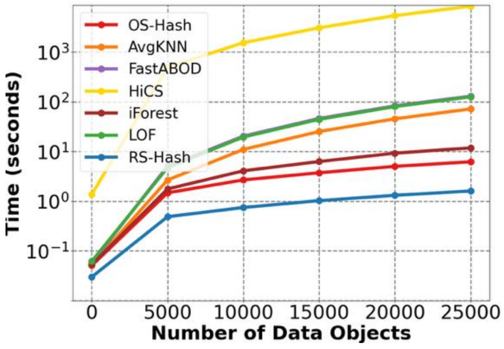
WAVEFORM



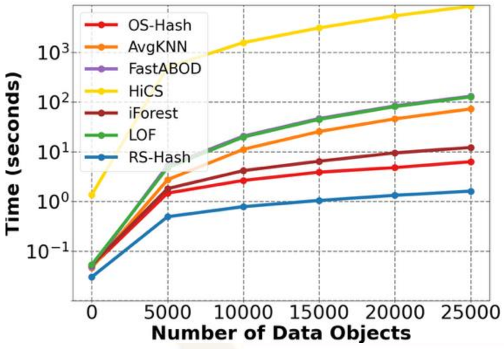
KDDCUP99



KDDCUP99



NORMAL(0,1)



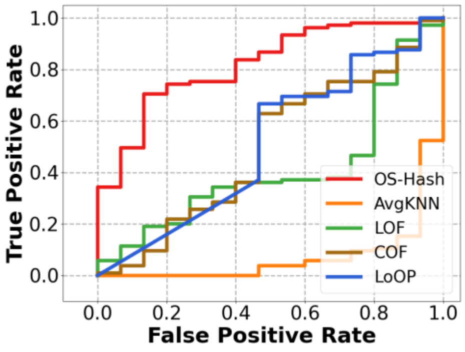
UNIFORM(0,1)

Static: Time Series Datasets

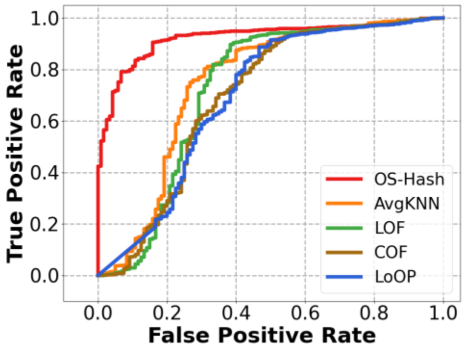
Dataset	OS-Hash	AvgKNN	LOF	COF	LoOP
PICKUP	**75.00**	74.50	73.00	68.50	63.75
PEBBLE	**79.17**	77.11	41.02	49.59	51.14
POWER	**66.03**	52.10	37.43	40.82	38.89
ECG5000	**92.17**	74.41	72.14	69.26	69.35
CROP	**83.96**	66.82	35.68	38.32	36.78

Table 5: AUC results for time series datasets

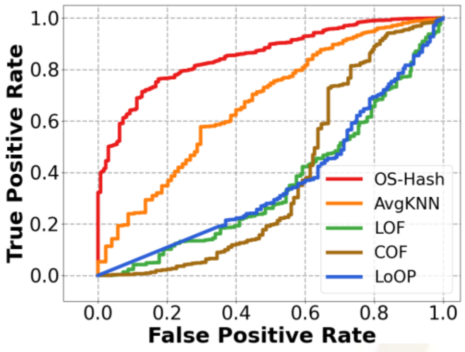
Static: Time Series Datasets



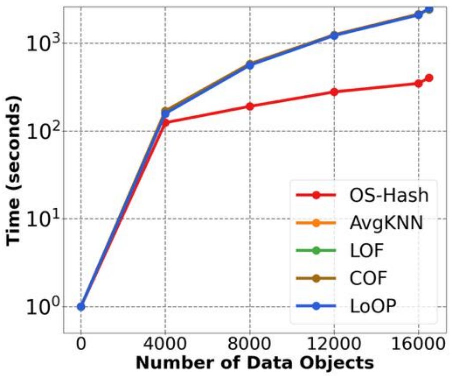
PEBBLE



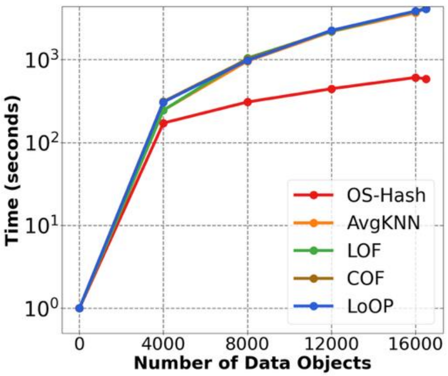
ECG5000



CROP

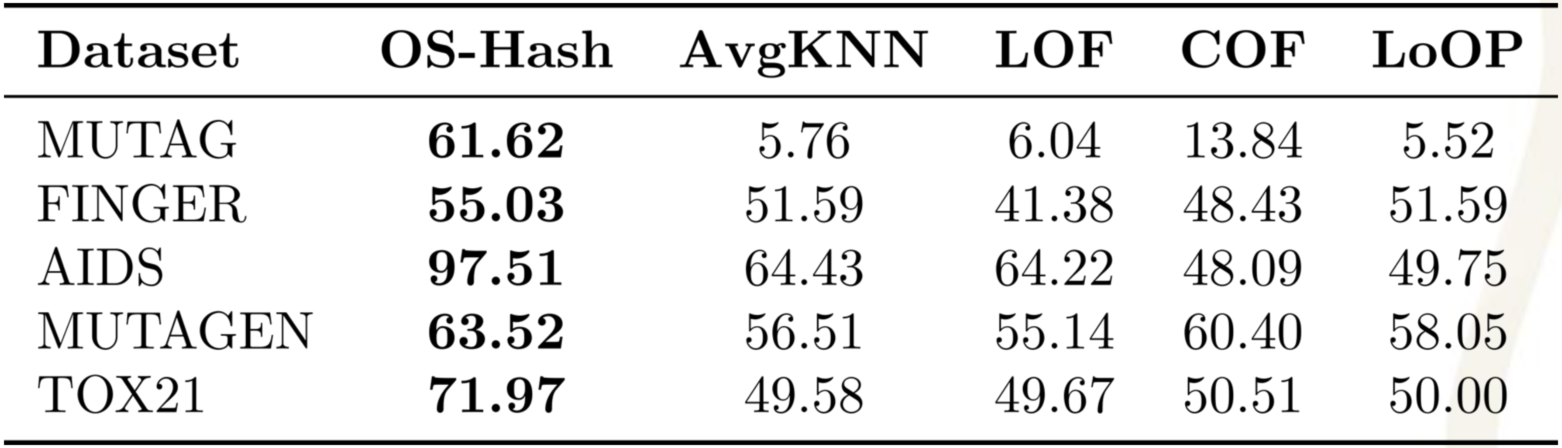


CROP



UNIFORM(0,1)

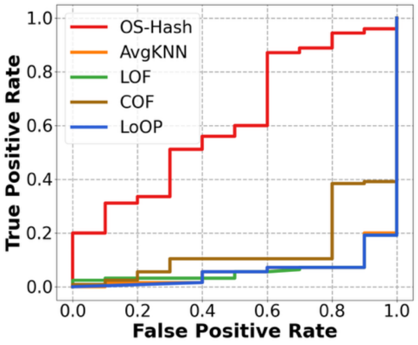
Static: Graph Datasets



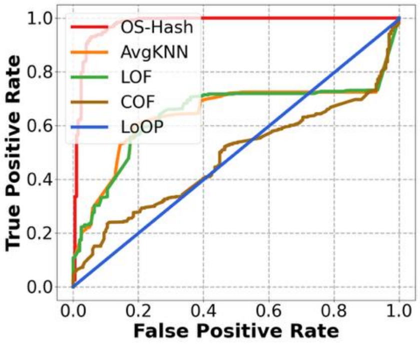
Dataset	OS-Hash	AvgKNN	LOF	COF	LoOP
MUTAG	**61.62**	5.76	6.04	13.84	5.52
FINGER	**55.03**	51.59	41.38	48.43	51.59
AIDS	**97.51**	64.43	64.22	48.09	49.75
MUTAGEN	**63.52**	56.51	55.14	60.40	58.05
TOX21	**71.97**	49.58	49.67	50.51	50.00

Table 6: AUC results for graph datasets

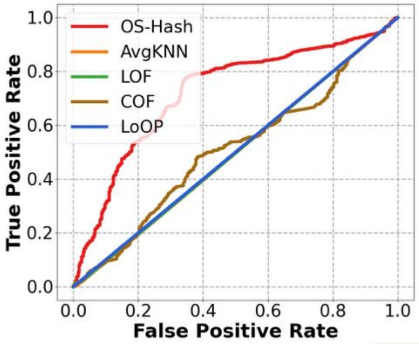
Static: Multidimensional Dataset



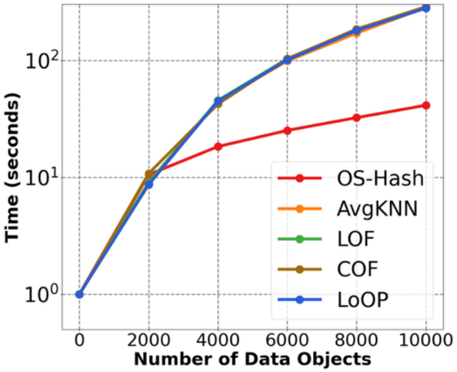
MUTAG



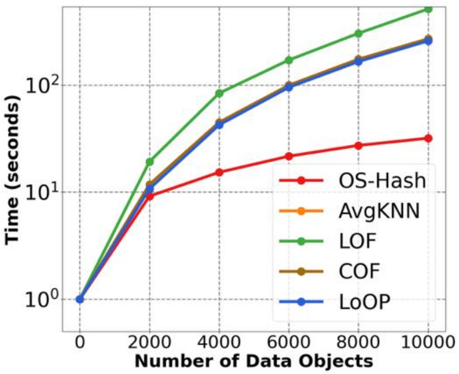
AIDS



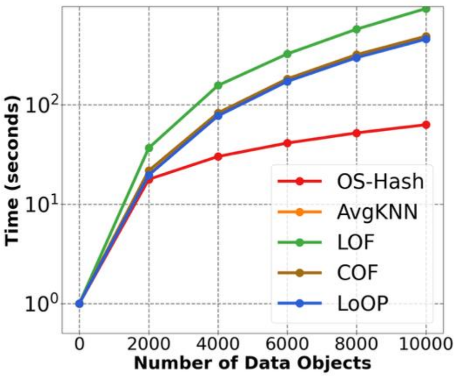
TOX21



TOX21

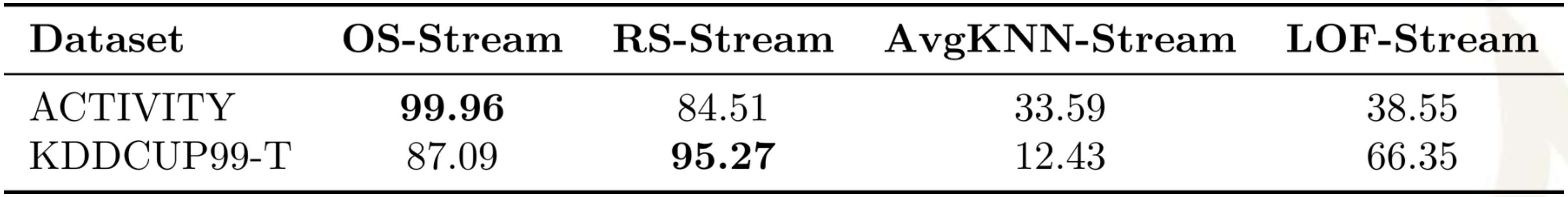


EDGE18



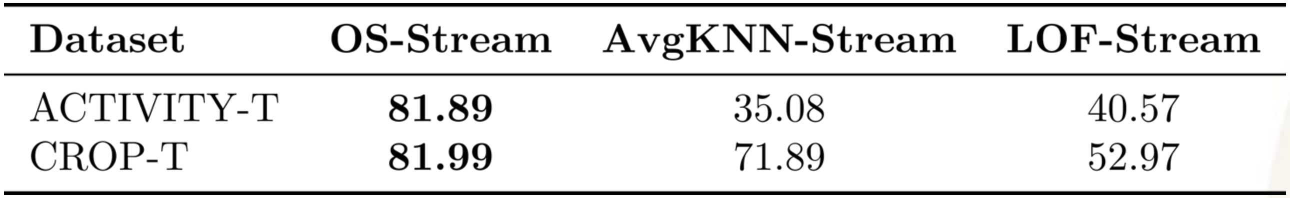
NODE18

Stream: AUC Scores



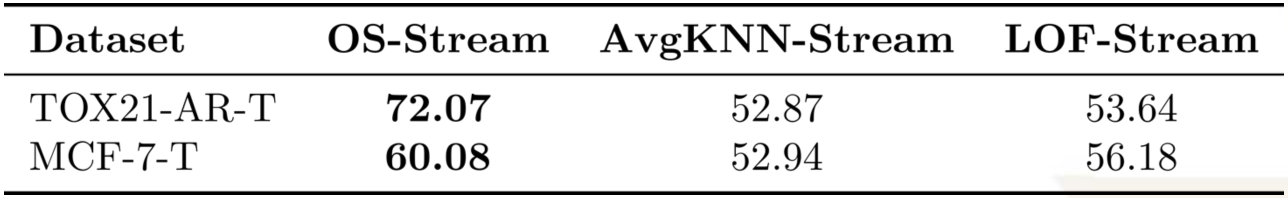
Dataset	OS-Stream	RS-Stream	AvgKNN-Stream	LOF-Stream
ACTIVITY	**99.96**	84.51	33.59	38.55
KDDCUP99-T	87.09	**95.27**	12.43	66.35

Table 7: AUC results in multidimensional data streams



Dataset	OS-Stream	AvgKNN-Stream	LOF-Stream
ACTIVITY-T	**81.89**	35.08	40.57
CROP-T	**81.99**	71.89	52.97

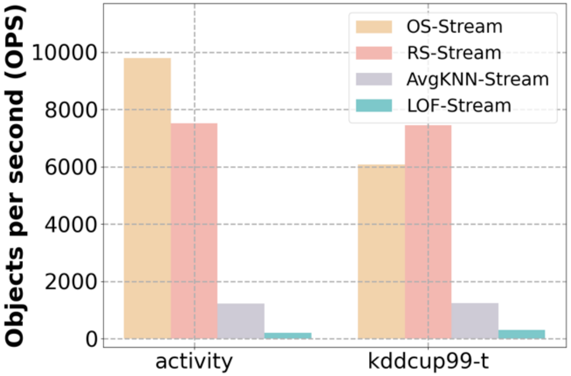
Table 8: AUC results in time series data streams



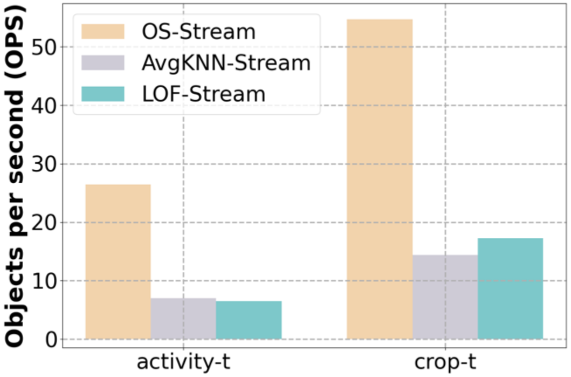
Dataset	OS-Stream	AvgKNN-Stream	LOF-Stream
TOX21-AR-T	**72.07**	52.87	53.64
MCF-7-T	**60.08**	52.94	56.18

Table 9: AUC results in graph data streams

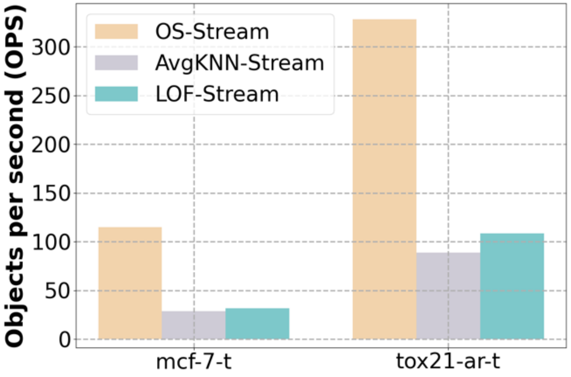
Stream: Object per Second



Multidimensional

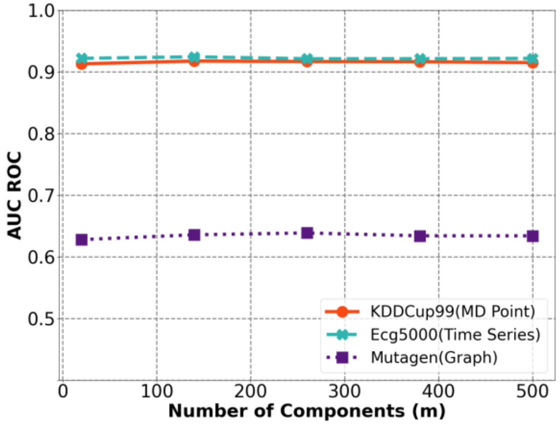


Time Series

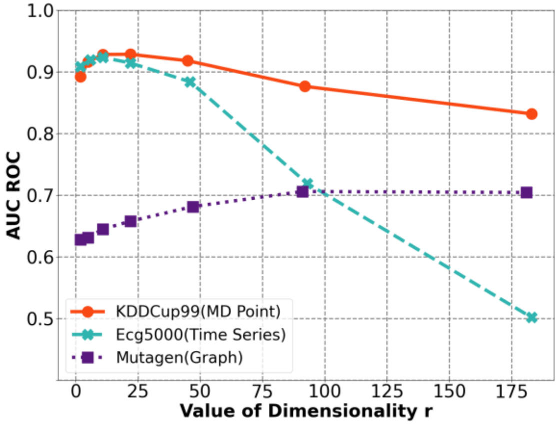


Graphs

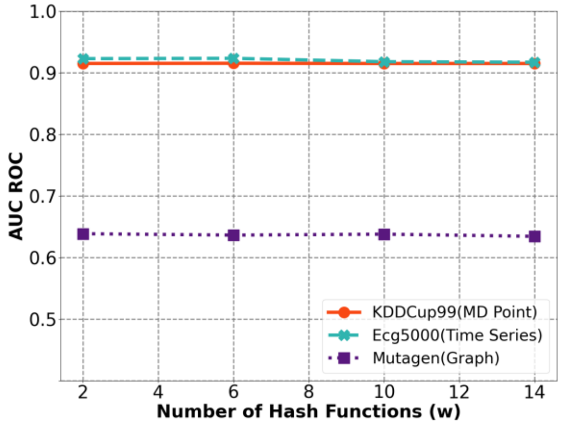
Parameter Analysis



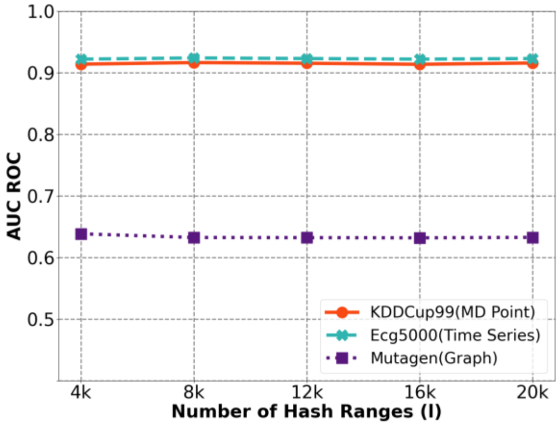
# Base Detectors, m



Dimensionality, r



# Hash Functions, w



Hash Table Size, l

Conclusion

Bowen Li

Seminars of Scientific Computation

Conclusion

- Proposed OS-Hash, a linear-time, constant-space oblique subspace outlier detector.
- Introduced oblique subspaces for arbitrary-shaped data.
- Applicable to multidimensional, time-series, graph, and streaming data.
- Experiments show superior accuracy, efficiency, and generality over state-of-the-art methods.

Thank You!