MStream: Fast Anomaly Detection in Multi-Aspect Streams (WWW 2021)

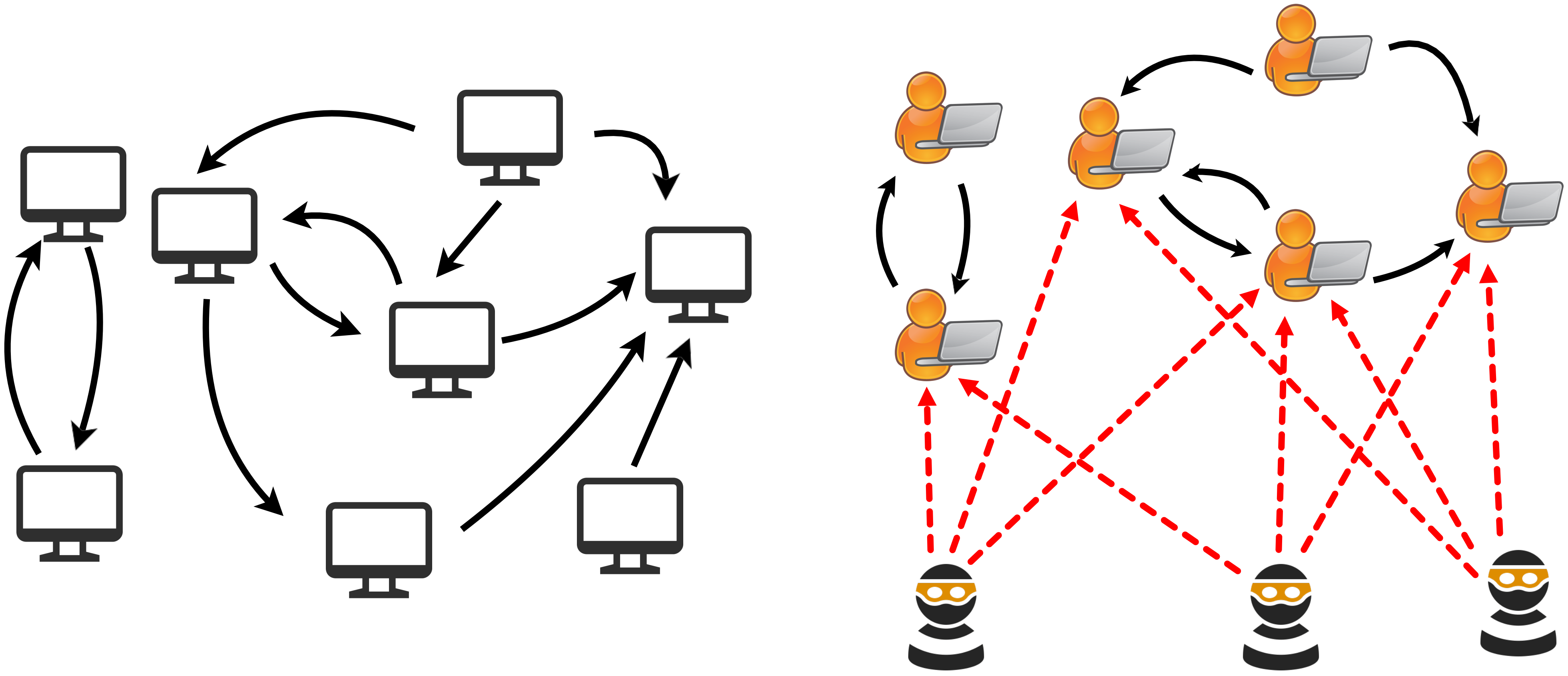
Siddharth Bhatia, Arjit Jain, Pan Li, Ritesh Kumar, Bryan Hooi

siddharth@comp.nus.edu.sg

@siddharthb_

https://github.com/Stream-AD/MStream/

Motivation



Roadmap

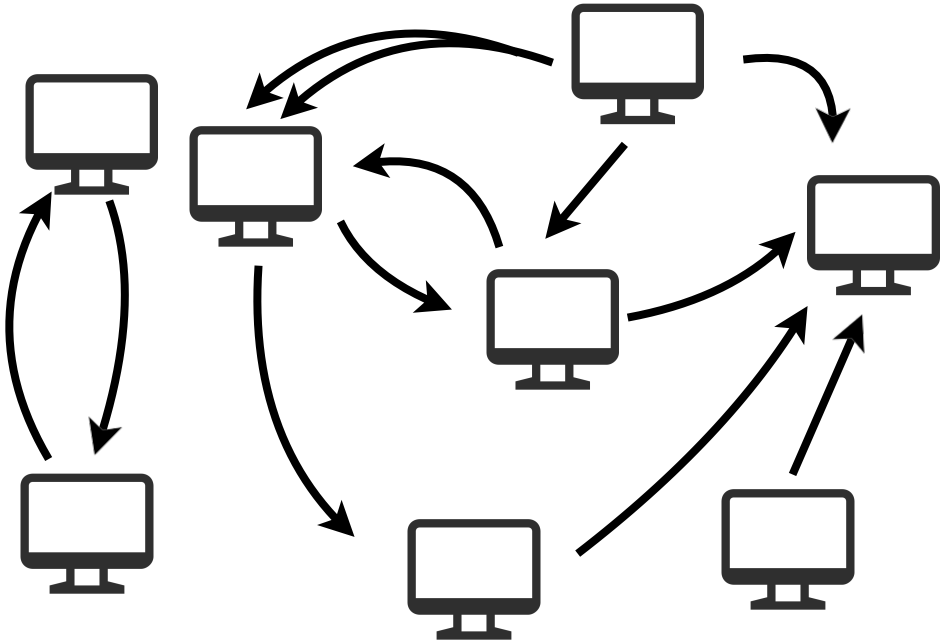
- **Problem**
- Algorithm
- Related Work
- Experiments
- Future Work

Problem

Input:
- Record stream $\boldsymbol{R}$
- Each having $\boldsymbol{d}$ dimensions

Output:
- Anomaly Score for each Record

Our Contributions:
- Multi-Aspect Group Anomaly Detection
- Streaming Approach
- Capture Correlation Between Features



Problem

Time	Source IP	Dest. IP	Pkt. Size	⋯
1	194.027.251.021	194.027.251.021	100	⋯
2	172.016.113.105	207.230.054.203	80	⋯
4	194.027.251.021	192.168.001.001	1000	⋯
4	194.027.251.021	192.168.001.001	995	⋯
4	194.027.251.021	192.168.001.001	1000	⋯
5	194.027.251.021	192.168.001.001	990	⋯
5	194.027.251.021	194.027.251.021	1000	⋯
5	194.027.251.021	194.027.251.021	995	⋯
6	194.027.251.021	194.027.251.021	100	⋯
7	172.016.113.105	207.230.054.203	80	⋯

Roadmap

- Problem
- **Algorithm**
- Related Work
- Experiments
- Future Work

MIDAS: CMS+Chi-squared test

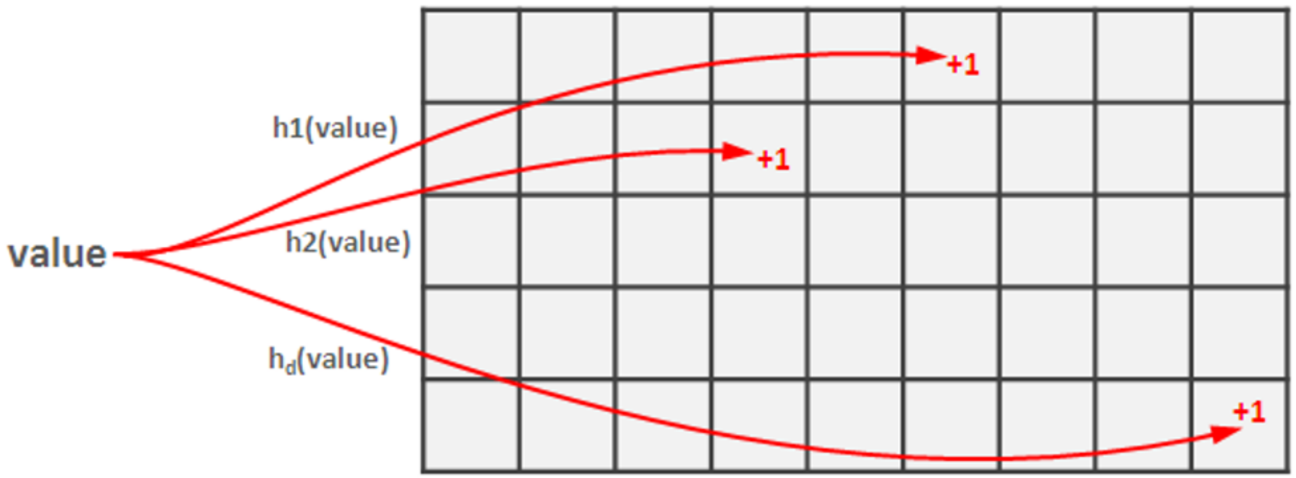
Background

$\hat{a}_{uv} \leq a_{uv} + \upsilon N_t$ with probability at least $1 - \varepsilon$
υ is the amount of error we can tolerate.
$1 - \varepsilon$ is the probability.
e.g. with 99% probability only up to 0.5% error

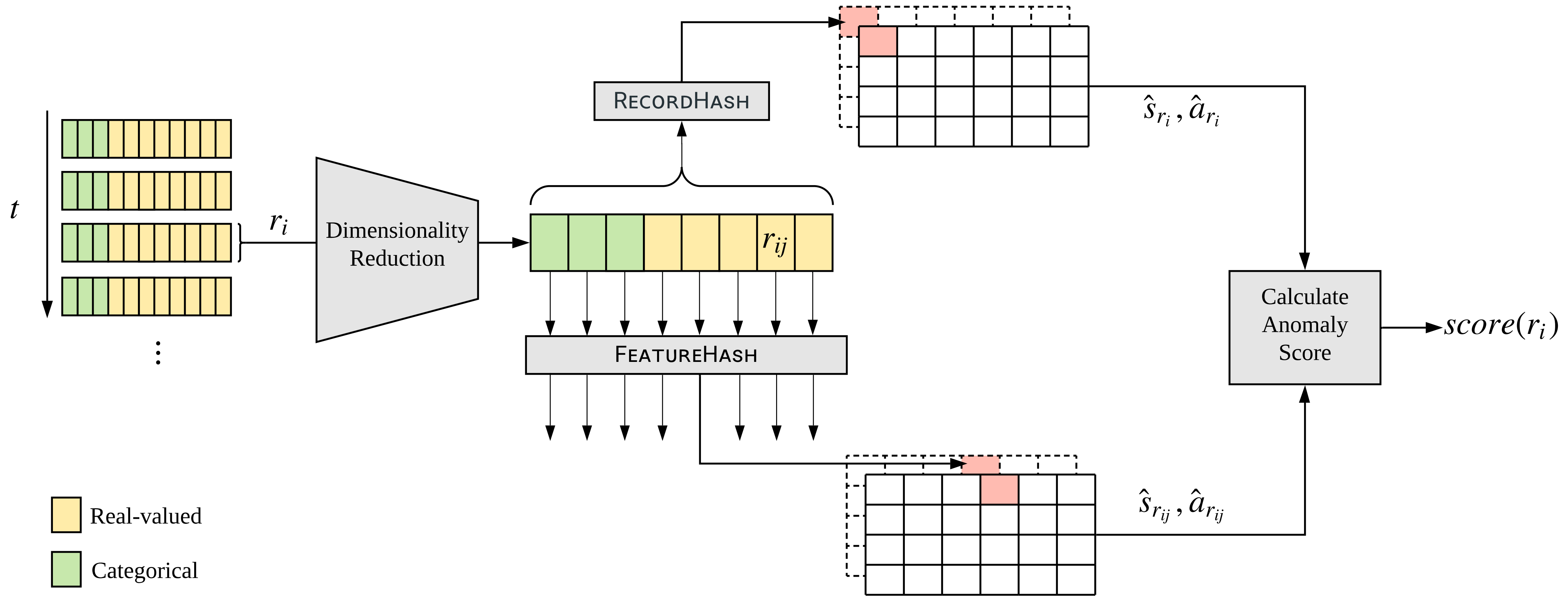
s_{uv} : $u - v$ edges up to time t
a_{uv} : $u - v$ edges at current time t

$\hat{s}_{uv}$: Approximate total count
$\hat{a}_{uv}$: Approximate current count

$$score((u,v,t)) = (\hat{a_{uv}} - \frac{\hat{s_{uv}}}{t})^2 \frac{t^2}{\hat{s_{uv}}(t-1)}$$



Algorithm



t
r_i
Dimensionality Reduction
RecordHash
r_{ij}
FeatureHash
$\hat{s}_{r_i}, \hat{a}_{r_i}$
$\hat{s}_{r_{ij}}, \hat{a}_{r_{ij}}$
Calculate Anomaly Score
$score(r_i)$
Real-valued
Categorical

Algorithm

Algorithm 1: FEATUREHASH: Hashing Individual Feature

Input: r_{ij} (Feature j of record r_i)
Output: Bucket index in $\{0, \ldots, b-1\}$ to map r_{ij} into

if r_{ij} is categorical
  output HASH(r_{ij}) `// Linear Hash [71]`
else if r_{ij} is real-valued
  ▷ **Log-Transform**
    $\tilde{r}_{ij} = \log(1 + r_{ij})$
  ▷ **Normalize**
    $\tilde{r}_{ij} \leftarrow \frac{\tilde{r}_{ij} - min_j}{max_j - min_j}$ `// Streaming Min-Max`
  output $\lfloor \tilde{r}_{ij} \cdot b \rfloor (\text{mod } b)$ `// Bucketization into` b `buckets`

Algorithm

Algorithm 2: RECORDHASH: Hashing Entire Record

Input: Record r_i
Output: Bucket index in $\{0, \dots, b-1\}$ to map r_i into
▷ **Divide r_i into its categorical part, r_i^{cat}, and its numerical part, r_i^{num}**
▷ **Hashing r_i^{cat}**
  $bucket_{cat} = (\sum_{j \in C} \text{HASH}(r_{ij})) \pmod b$ `// Linear Hash [71]`
▷ **Hashing r_i^{num}**
  for $id \leftarrow 1$ to k
    if $\langle r_i^{num}, \mathbf{a}_{id} \rangle > 0$
      $bitset[id] = 1$
    else
      $bitset[id] = 0$
  $bucket_{num} = \text{INT}(bitset)$ `// Convert bitset to integer`
output $(bucket_{cat} + bucket_{num}) \pmod b$

Algorithm

Algorithm 3: MSTREAM: Streaming Anomaly Scoring

Input: Stream of records over time
Output: Anomaly scores for each record

```
▹ Initialize data structures:
    Total record count ŝ_{r_i} and total attribute count ŝ_{r_ij} ∀j ∈ {1, .., d}
    Current record count â_{r_i} and current attribute count â_{r_ij} ∀j ∈ {1, .., d}
while new record (r_i, t) = (r_i1, ..., r_id, t) is received: do
    ▹ Hash and Update Counts:
      for j ← 1 to d
          bucket_j = FEATUREHASH(r_ij)
          Update count of bucket_j
      bucket = RECORDHASH(r_i)
      Update count of bucket
    ▹ Query Counts:
      Retrieve updated counts ŝ_{r_i}, â_{r_i}, ŝ_{r_ij} and â_{r_ij} ∀j ∈ {1..d}
    ▹ Anomaly Score:
      output
```

$$score(r_i, t) = \left(\hat{a}_{r_i} - \frac{\hat{s}_{r_i}}{t}\right)^2 \frac{t^2}{\hat{s}_{r_i}(t-1)} + \sum_{j=1}^{d} score(r_{ij}, t)$$

Incorporating Correlation Between Features

1. Principal Component Analysis

2. Information Bottleneck

3. Autoencoder

Time and Memory Complexity

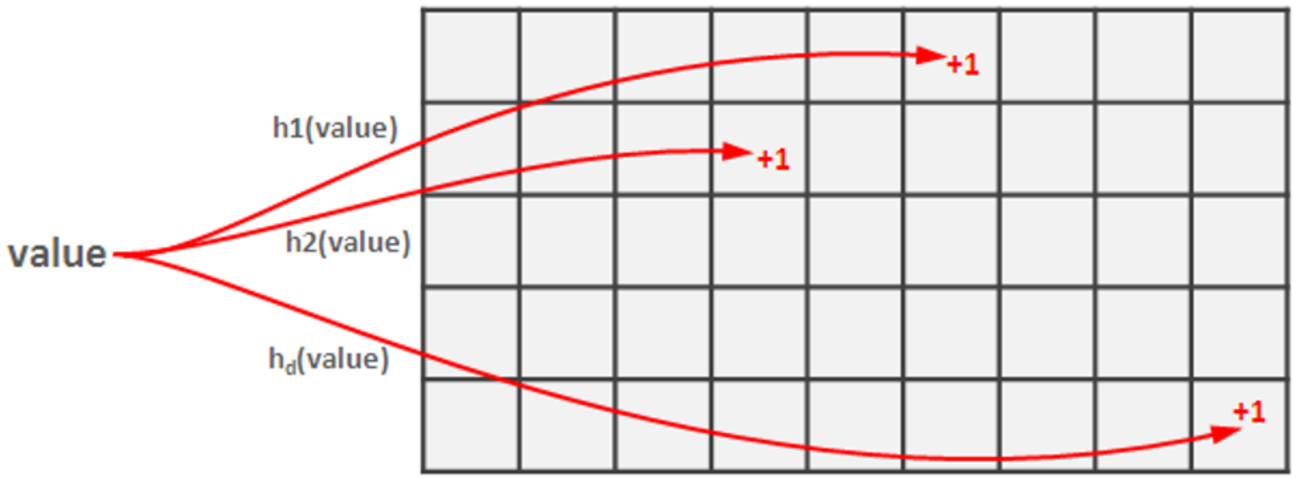
w: number of hash functions
b: number of buckets
d: number of dimensions/features

Space complexity:

- $O(wbd)$

Time complexity:

- $O(wd)$



Roadmap

- Problem
- Algorithm
- **Related Work**
- Experiments
- Future Work

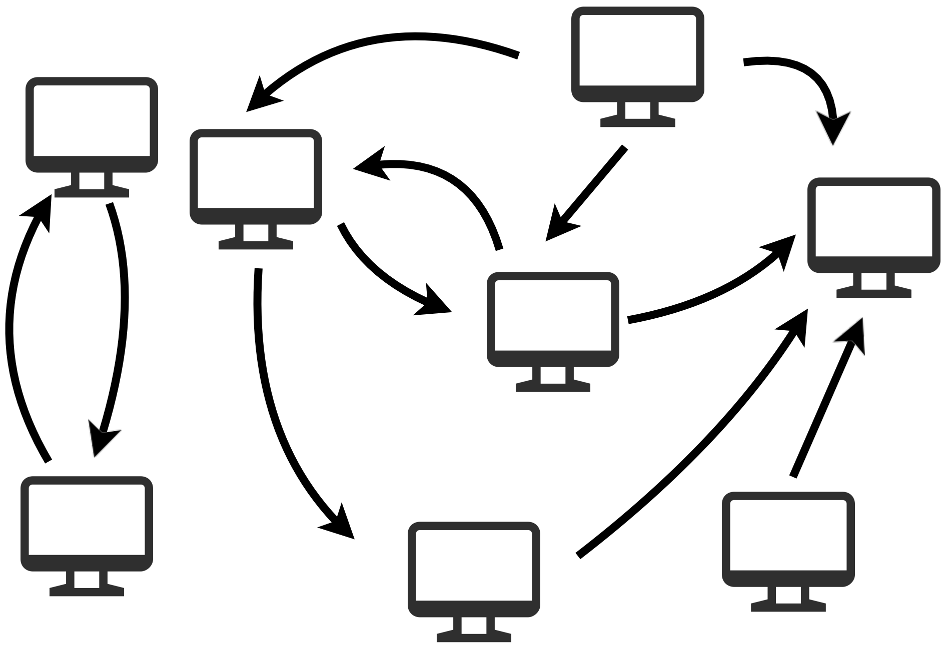
Related Work

	Elliptic (1999)	LOF (2000)	I-Forest (2008)	STA (2006)	MASTA (2015)	STenSr (2015)	Random Cut Forest (2016)	DENSEALERT (2017)	**MSTREAM (2021)**
Group Anomalies								✓	✔
Real-valued Features	✓	✓	✓				✓		✔
Constant Memory							✓	✓	✔
Const. Update Time				✓	✓	✓	✓	✓	✔

Roadmap

- Problem
- Algorithm
- Related Work
- **Experiments**
- Future Work

Datasets



1. *KDDCUP99*: 1.21M records (20% anomalies), 42 features
2. CICIDS-DoS: 1.05M records (5% anomalies), 80 features
3. UNSW-NB15: 2.5M records (13% anomalies), 49 features
4. CICIDS-DDoS: 7.9M records (7% anomalies), 83 features

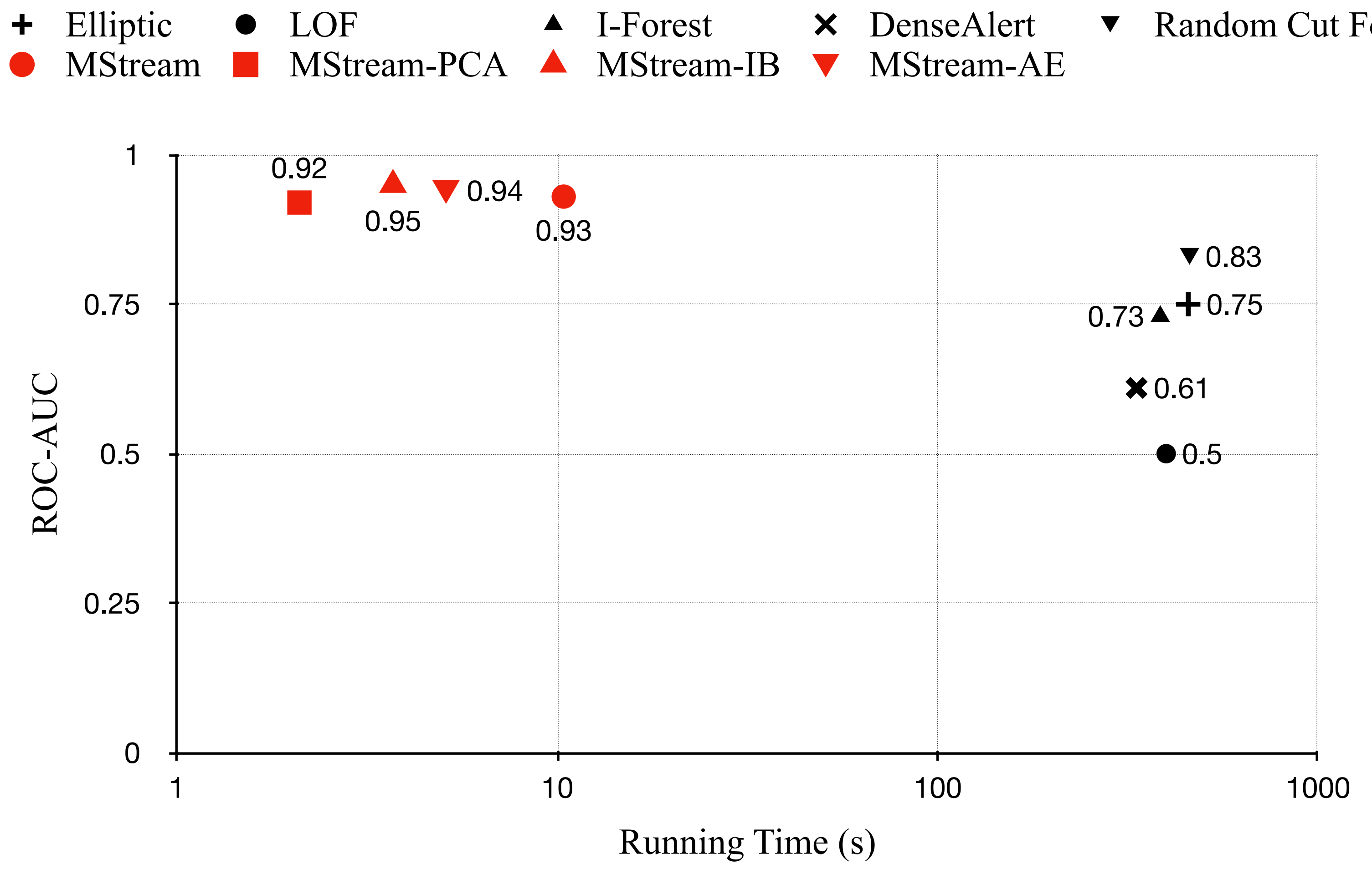
Area under the ROC curve (AUC)

	Elliptic	LOF	I-Forest	DAlert	RCF	MStream	MStream-PCA	MStream-IB	MStream-AE
KDD	0.34 ± 0.025	0.34	0.81 ± 0.018	0.92	0.63	0.91 ± 0.016	0.92 ± 0.000	**0.96** ± 0.002	**0.96** ± 0.005
DoS	0.75 ± 0.021	0.50	0.73 ± 0.008	0.61	0.83	0.93 ± 0.001	0.92 ± 0.001	**0.95** ± 0.003	0.94 ± 0.001
UNSW	0.25 ± 0.003	0.49	0.84 ± 0.023	0.80	0.45	0.86 ± 0.001	0.81 ± 0.001	0.82 ± 0.001	**0.90** ± 0.001
DDoS	0.57 ± 0.106	0.46	0.56 ± 0.021	−−	0.63	0.91 ± 0.000	**0.94** ± 0.000	0.82 ± 0.000	0.93 ± 0.000

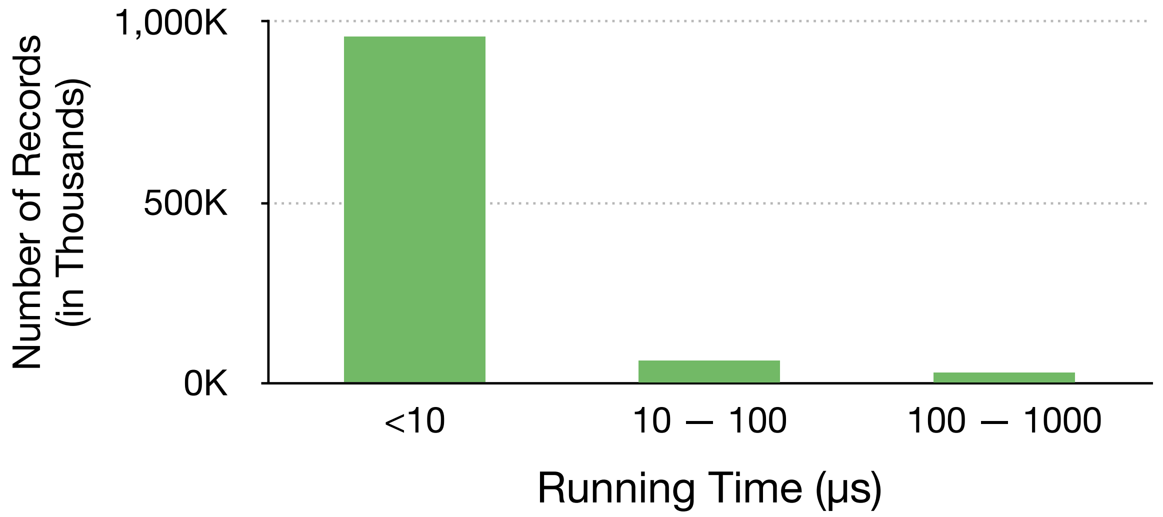
Running Times

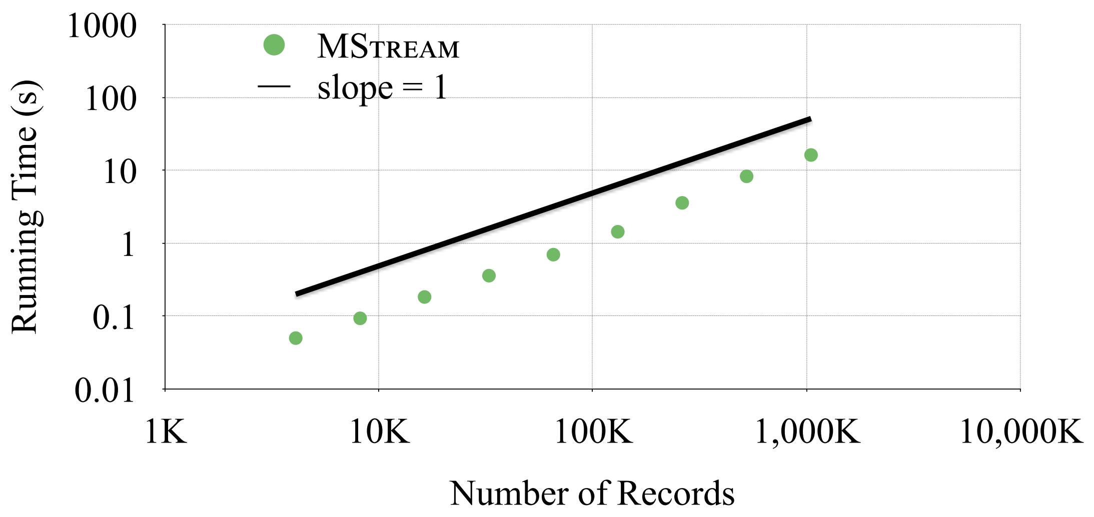
	Elliptic	LOF	I-Forest	DAlert	RCF	MStream	MStream-PCA	MStream-IB	MStream-AE
KDD	216.3	1478.8	230.4	341.8	181.6	4.3	2.5	3.1	3.1
DoS	455.8	398.8	384.8	333.4	459.4	10.4	2.1	3.7	5.1
UNSW	654.6	2091.1	627.4	329.6	683.8	12.8	6.6	8	8
DDoS	3371.4	15577*s*	3295.8	−−	4168.8	61.6	16.9	25.6	27.7

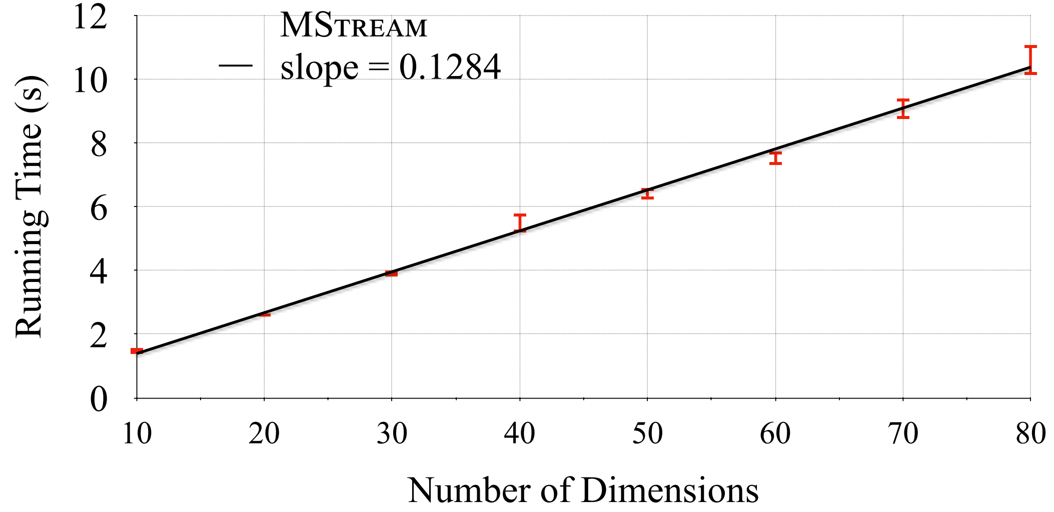
AUC vs Time

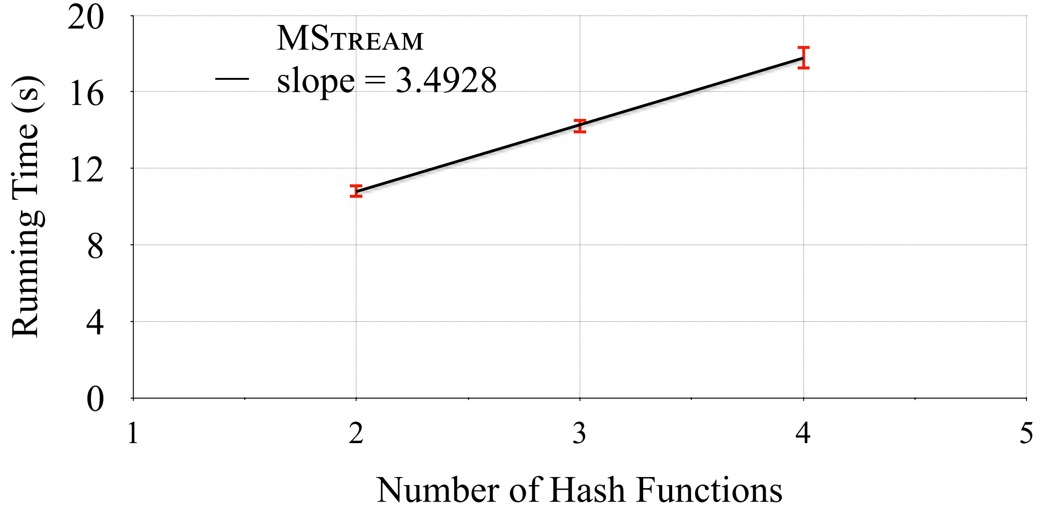

Elliptic
LOF
I-Forest
DenseAlert
Random Cut Forest
MStream
MStream-PCA
MStream-IB
MStream-AE
1
0.75
0.5
0.25
0
ROC-AUC
0.92
0.95
0.94
0.93
0.83
0.73
0.75
0.61
0.5
1
10
100
1000
Running Time (s)

Scalability

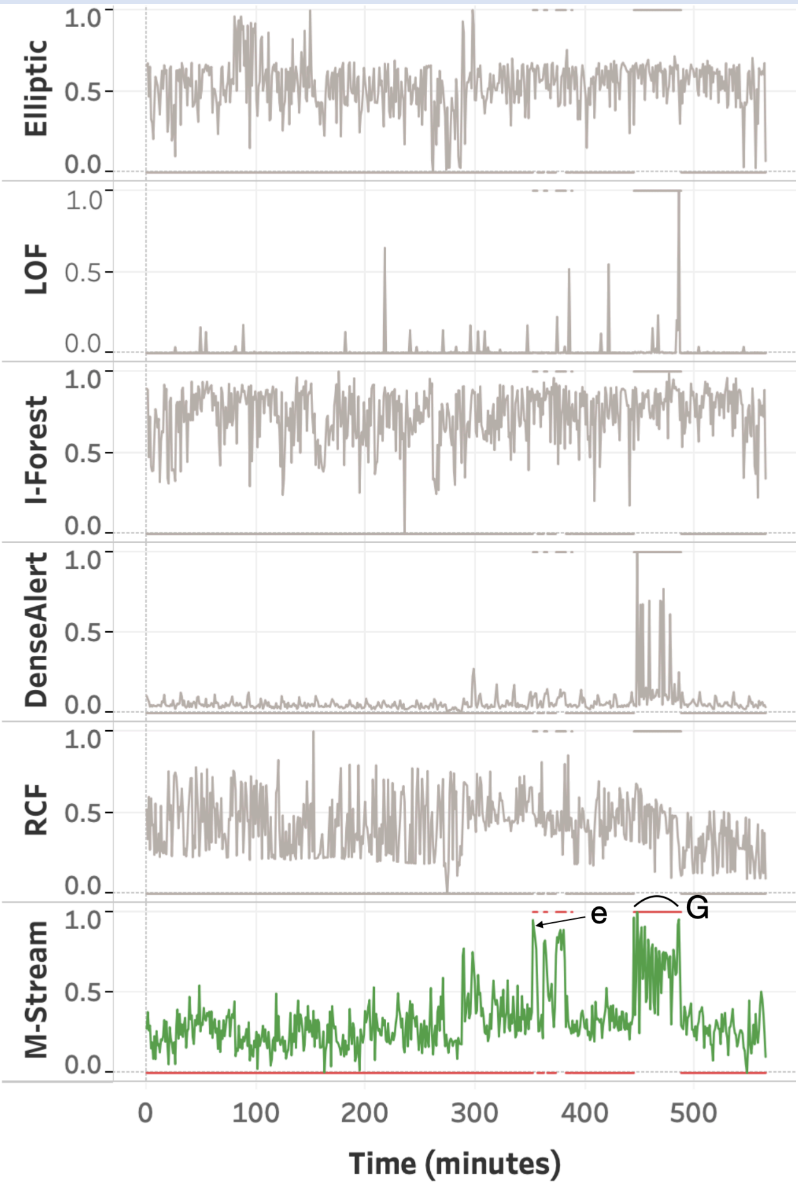








Discoveries



Roadmap

- Problem
- Algorithm
- Related Work
- Experiments
- **Future Work**

Future Work

1. Semi-Supervision
2. Few Labels
3. Generating Anomalous Data

Conclusion

1. Multi-Aspect Group Anomaly Detection:
   - Categorical and Numeric Attributes

2. Streaming Approach:
   - Constant Memory and Update Time

3. Effectiveness:
   - Capture Correlation Between Features

Siddharth Bhatia, Bryan Hooi, Minji Yoon, Kijung Shin and Christos Faloutsos. "MStream: Fast Anomaly Detection in Multi-Aspect Streams." The Web Conference (WWW), 2021. https://arxiv.org/abs/2009.08451

https://github.com/Stream-AD/MStream/