Anomaly Detection for River

DATAAI967 - Data Science in Practice

Maya Awada - Ariel Ramos

Introduction

- Adapting PySAD anomaly detection algorithms (XStream - LODA - RSHash) to River
    - Removal of Dependencies
    - Code Documentation
    - River Guidelines
- Benchmarking Experiments
- Comparison to the XStream paper results
- Demonstration using Docker Image

Real-life Applications

Online Anomaly Detection is useful in many applications:

- Detecting SPAM SMS
- Fraud Detection in Financial Transactions
- Intrusion Detection in Computer Network
- Monitoring Sensor Reading in an Aircraft
- Spotting Potential Risk or Medical Problem in Health Data

XStream

Density-based ensemble outlier detection algorithm

- It measures outlierness at multiple scales or granularities
- It can handle high-dimensionality through distance-preserving projections

Window-based approach

→ Bin counts accumulated in the previous window are used to score points in the current window, with windows sliding forward periodically after each current window is full.

XStream

Method Key Components:

1- StreamHash

Subspace-selection and dimensionality reduction via sparse random projections.

2- Half-Space Chains

An efficient ensemble to estimate density at multiple scales.

XStream: StreamHash

Streamhash Random Projection Method:

Each hash function h_i maps a string f (the feature name) to a hash-value, $h_i : f \to R$

$$h_i[f] = \sqrt{\frac{3}{K}} \begin{cases} -1 & \text{if } a_i(f) \in [0, 1/6) \\ 0 & \text{if } a_i(f) \in [1/6, 5/6) \\ +1 & \text{if } a_i(f) \in [5/6, 1] \end{cases}$$

with $a_i(f) = g_i(f) / (2^{32} - 1)$, in $[0, 1]$

Random Projection via StreamHash:

$$\boxed{\boldsymbol{y}[i] = \sum_{f_j \in \mathcal{F}} h_i(f_j)\, \boldsymbol{x}[j], \quad i = 1, \ldots, K.}$$

Projected point $\mathbf{y} \in R^k$

Point $\mathbf{x} \in R^d$

XStream: Half-Space Chains

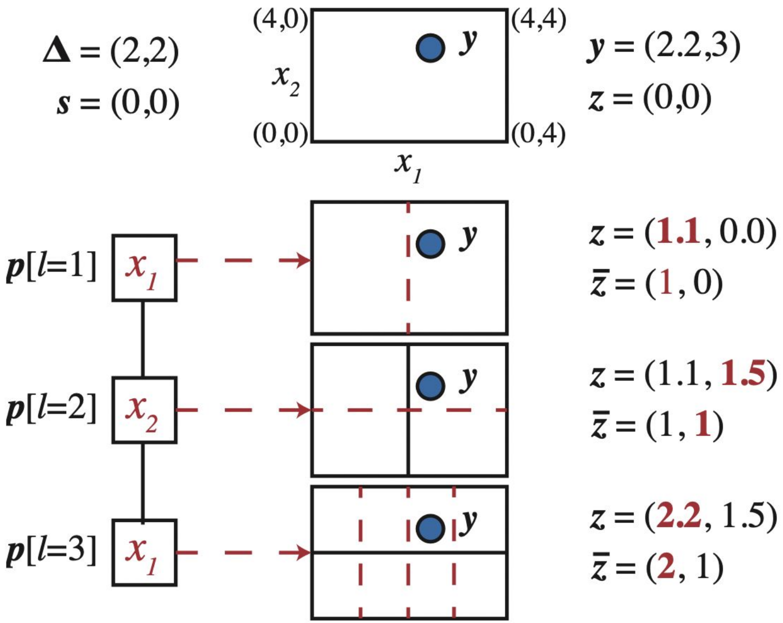
XStream is an ensemble of Half-Space Chains

Each chain randomly selects a single split-dimension $p \in P$ at each level $l = 1,\ldots, D$ of the chain, and recursively splits the space along that dimension into discrete bins.

XStream: Half-Space Chains

K = 2 dimensional projected space

Example of a half-space chain of depth D=3:



1.1 = 2.2/2

Z : Bin-vector

Note: The bin-vector of a point at a certain level identifies the bin in which the point falls at that level.

XStream: Code Documentation

```
class xStream:
    '''
    XStream Algorithm

    Parameters
    ----------
    streamhash
        StreamhashProjection class object.
    deltamax
        List of bin-widths corresponding to half the range of the projected data.
    window_size
        Number of points to observe before replacing the counts in the reference window by those of the current window.
    chains
        Chains class object.
    step
        Counter for the number of points observed.
    cur_window
        Bin-counts for the current window.
    ref_window
        Bin-counts for the reference window.
    '''
```

XStream: Code Documentation

```
class Chain:
    '''
    Individual Chain

    Method to estimate density at multiple scales
    The chain approximates the density of a point by counting its nearby neighbors at multiple scales.
    For every scale or level, a count-min-sketch approximates the bin-counts at that level.
    Non-stationarity of data is handled by maintaining separate bin-counts for an alternating pair of windows
    containing ψ points each, termed as current and reference windows.

    Parameters
    ----------
    k
        Number of components or random projections.
    deltamax
        List of bin-widths corresponding to half the range of the projected data.
    depth
        Number of feature splits to be performed. Set to 25 by default.
    fs
        List containing the randomly selected split features or dimensions.
    cmsketches_ref
        Reference count-min-sketches corresponding to the reference window.
    cmsketches_cur
        Current count-min-sketches corresponding to the current window.
    rand_arr
        List of uniform random numbers used to compute the shift values.
    shift
        List containing the uniform shift value for every component.
    is_first_window
        Boolean value indicating whether the window under consideration is the first one or not.
    '''
```

XStream: Code Documentation

```
class Chains:

    '''
    Ensemble of Chains

    Parameters
    ----------
    n_chains
        Number of chains in the ensemble. Set to 100 by default.
    depth
        Number of feature splits to be performed. Set to 25 by default.
    chains
        Array grouping all the chains.

    '''
```

XStream: Code Documentation

```
class StreamhashProjection:
    '''
    Streamhash Projection.

    Method for subspace-selection and dimensionality reduction via sparse random projections.
    It reduces data dimensionality while accurately preserving distances between points,
    which facilitates outliers detection.

    Parameters
    ----------
    keys
        Array containing the indexes of the random projections.
    constant
        Constant value used in the hash value computation.
    density
        Fraction of non-zero components in the random projections. Set to 1/3.0 by default.
    n_components
        Number of random projections.
    seed
        Random number seed.
    '''
```

XStream Paper Results

Static Datasets Experiments

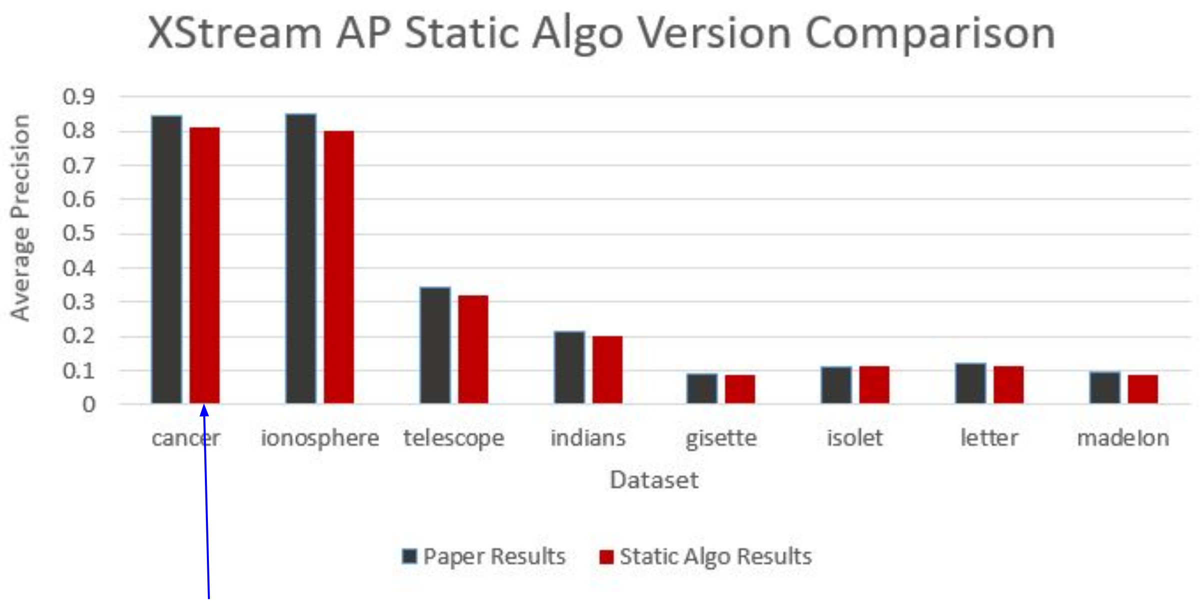
Dataset	*iForest*	*HS-Trees*	*RS-Hash*	*LODA*	xStream
cancer	0.617 ± 0.021	0.646 ± 0.033	0.619 ± 0.030	0.826 ± 0.013	0.845 ± 0.008
ionosphere	0.705 ± 0.006	0.706 ± 0.007	0.764 ± 0.032	0.642 ± 0.067	0.848 ± 0.018
telescope	0.367 ± 0.008	0.392 ± 0.012	0.391 ± 0.012	0.322 ± 0.007	0.344 ± 0.009
indians	0.142 ± 0.003	0.146 ± 0.002	0.156 ± 0.007	0.177 ± 0.008	0.216 ± 0.010
gisette	0.078 ± 0.002	0.080 ± 0.002	0.084 ± 0.007	0.087 ± 0.003	0.090 ± 0.003
isolet	0.099 ± 0.003	0.097 ± 0.005	0.108 ± 0.004	0.089 ± 0.004	0.112 ± 0.006
letter	0.093 ± 0.001	0.092 ± 0.002	0.104 ± 0.004	0.094 ± 0.006	0.122 ± 0.005
madelon	0.110 ± 0.003	0.101 ± 0.013	0.092 ± 0.005	0.101 ± 0.010	0.097 ± 0.004
Avg Rank	3.75	3.3125	2.875	3.3125	1.75

Average Precision on static datasets

Note:

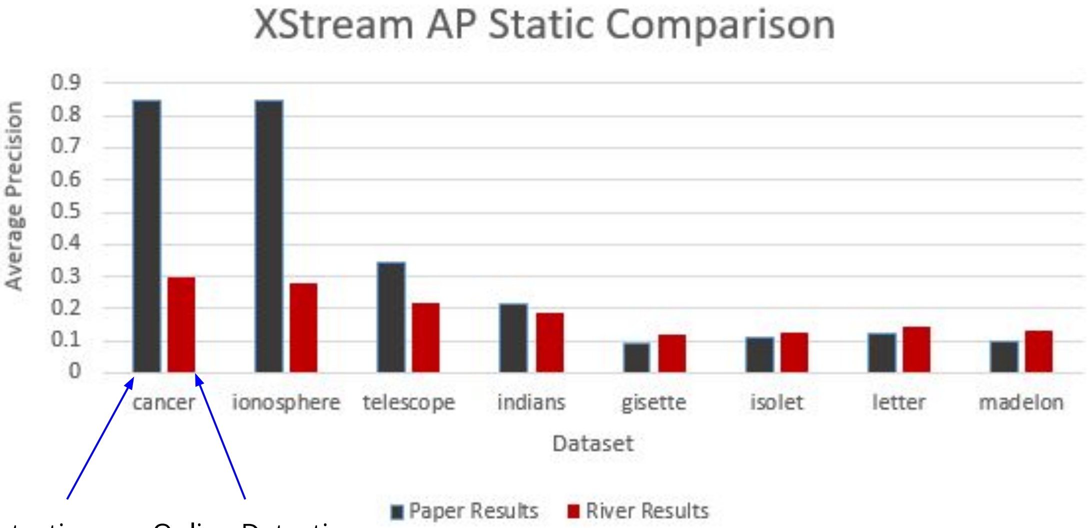
- These experiments are performed on the static version of the algorithms
- Offline Anomaly Detection: Static data inputted as one batch

XStream Experiments – Static Datasets



Static Version of the algorithm

XStream Experiments – Static Datasets



Offline Detection Online Detection

→ **Results not really comparable**

XStream Paper Results

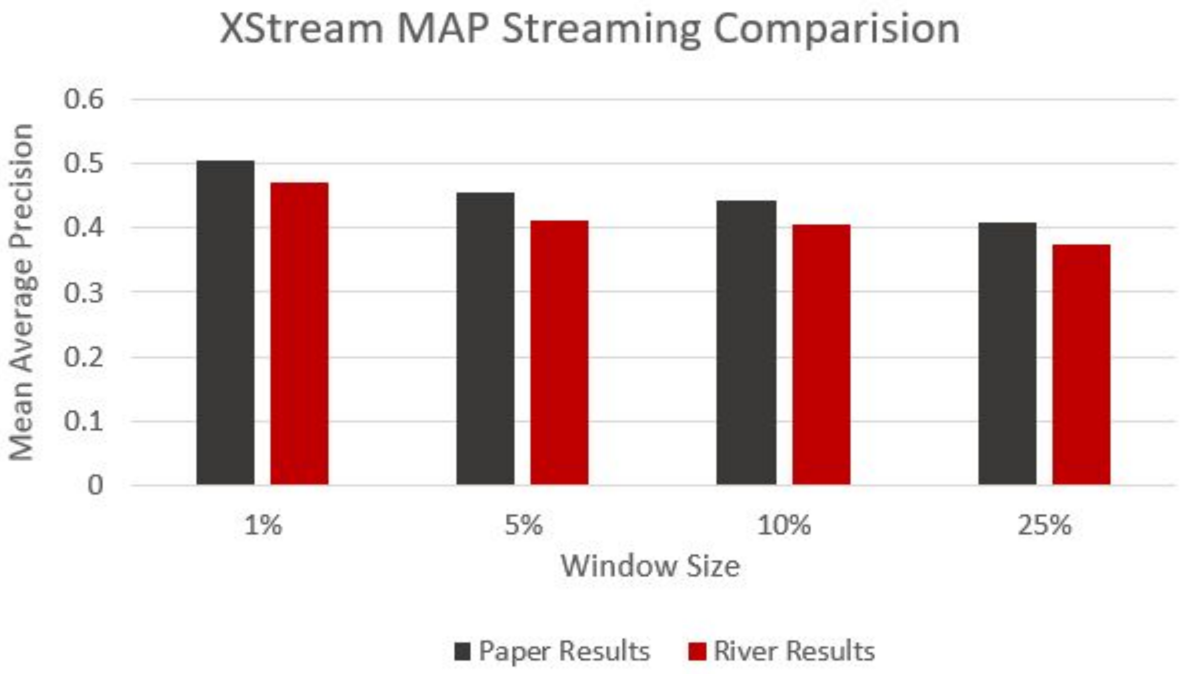
Streaming Dataset Experiments: SPAM-SMS Dataset

Window size ψ	***HS-Stream***		***LODA***		***RS-Hash***		xStream		xStream-**1K**	
	MAP	*OAP*	*MAP*	*OAP*	*MAP*	*OAP*	*MAP*	*OAP*	*MAP*	*OAP*
1%	0.480 ± 0.178	0.416	0.090 ± 0.028	0.076	0.291 ± 0.129	0.171	0.505 ± 0.138	0.422	0.522 ± 0.153	0.430
5%	0.492 ± 0.179	0.416	0.082 ± 0.014	0.077	0.216 ± 0.034	0.195	0.455 ± 0.135	0.406	0.493 ± 0.134	0.415
10%	0.430 ± 0.024	0.419	0.081 ± 0.010	0.080	0.174 ± 0.017	0.164	0.444 ± 0.037	0.433	0.448 ± 0.037	0.436
25%	0.363 ± 0.024	0.359	0.080 ± 0.001	0.080	0.203 ± 0.014	0.201	0.409 ± 0.009	0.404	0.435 ± 0.013	0.429

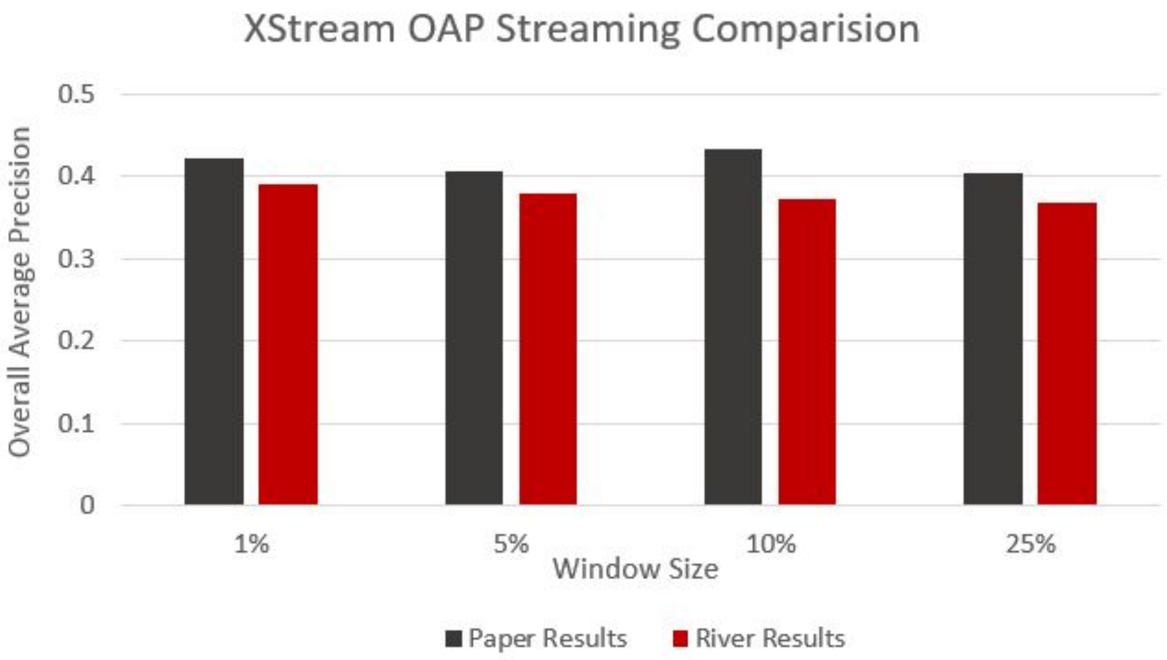
Evaluation Metrics

- Mean Average Precision (MAP)
- Overall Average Precision (OAP)

XStream Experiments – MAP Streaming



XStream Experiments – OAP Streaming



LODA

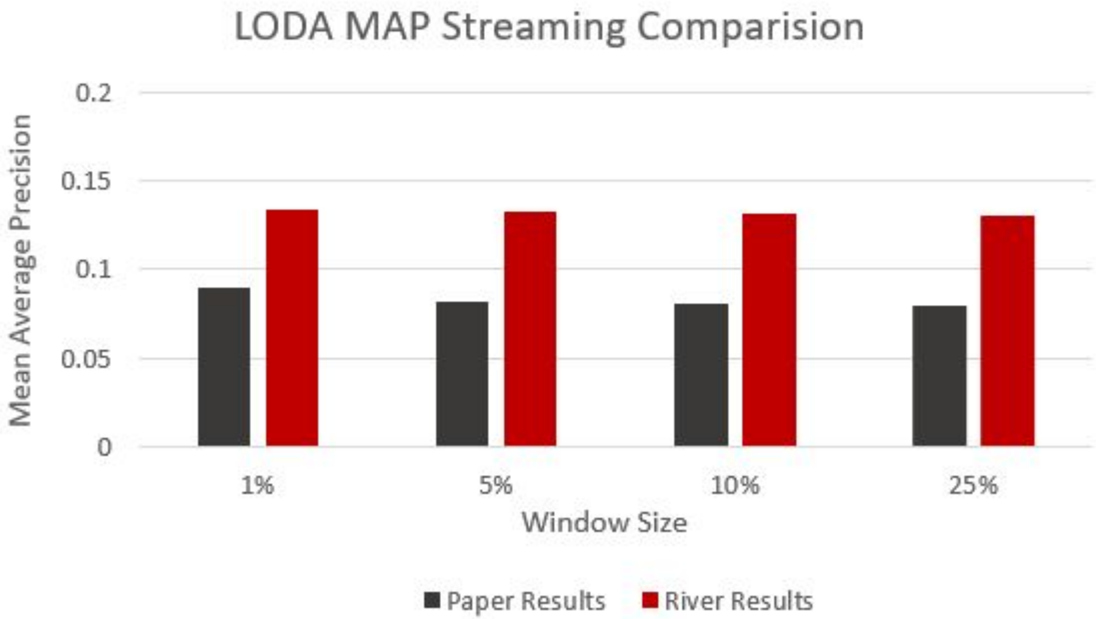
LODA = **L**ightweight **O**nline **D**etector of **A**nomalies

- Loda is composed of an ensemble of one-dimensional histograms
- Each histogram approximates the probability density of input data projected onto a single projection vector

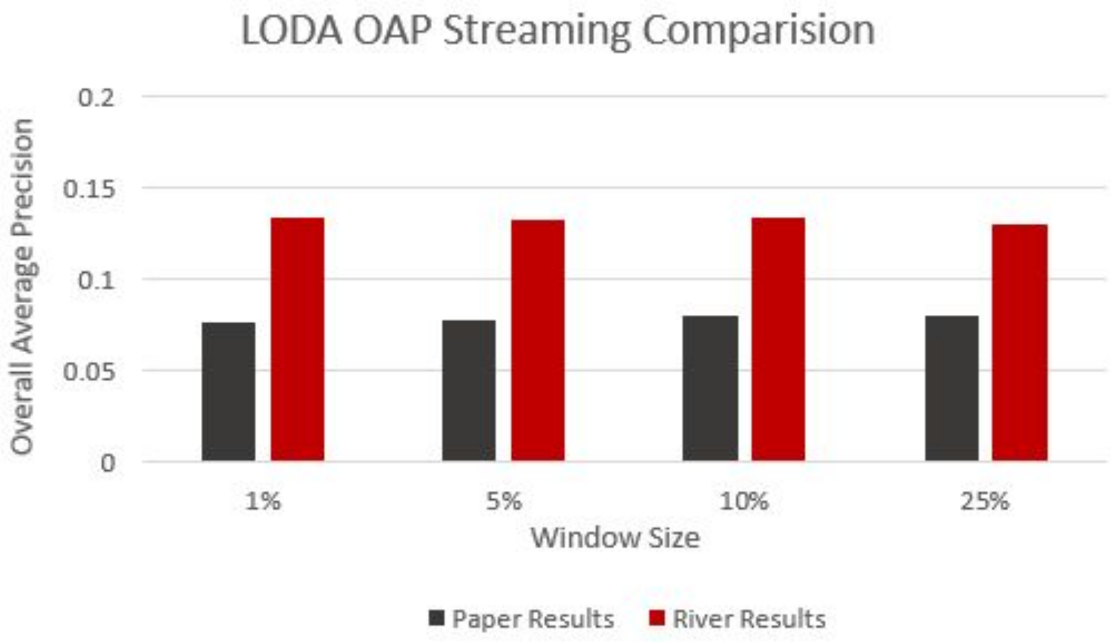
LODA: Code Documentation

```
class LODA():
    """
    Lightweight Online Detector of Anomalies

    Outlier detection algorithm that computes the likelihood of
    data points using an ensemble of one-dimensional histograms.

    Parameters
    ----------
    num_bins
        Number of bins of the histogram.
    num_random_cuts
        Number of random cut projections.
    """
```

LODA Experiments – MAP Streaming



LODA Experiments – OAP Streaming



RSHash

RSHash = **R**andomized **S**ubspace **Hash**ing algorithm

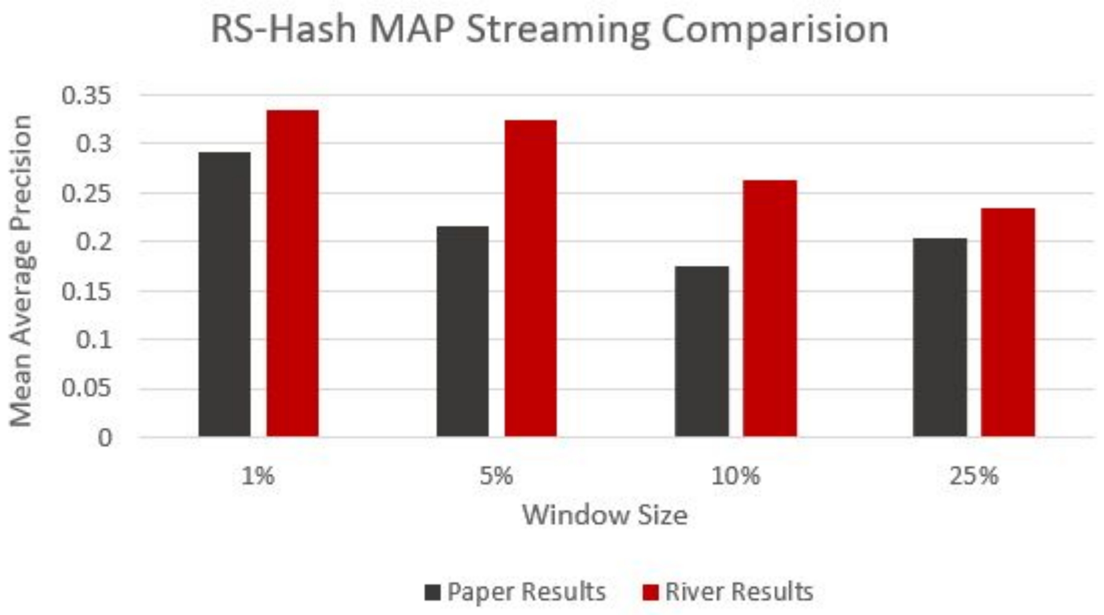
Outlier detector that relies on randomized hashing.

- Creates a hashed representation of the data
- Computes the log-likelihood density model using time-decayed scores

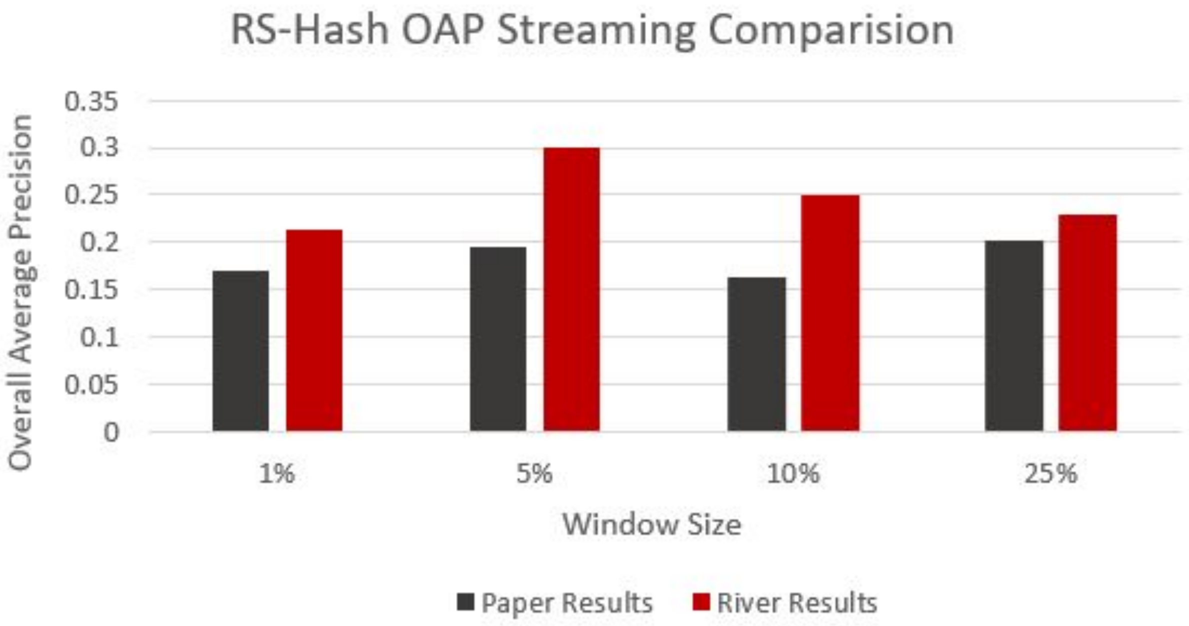
RSHash: Code Documentation

```python
class RSHash():
    """
    RSHash Algorithm

    Subspace outlier detector based on randomized hashing.

    Parameters
    ----------
    feature_mins
        Minimum boundary of the features.
    feature_maxes
        Maximum boundary of the features.
    sampling_points
        Number of sampling points.
    decay
        Decay hyperparameter.
    num_components
        Number of ensemble components.
    num_hash_fns
        Number of hashing functions.
    """
```

RS-Hash Experiments – MAP Streaming

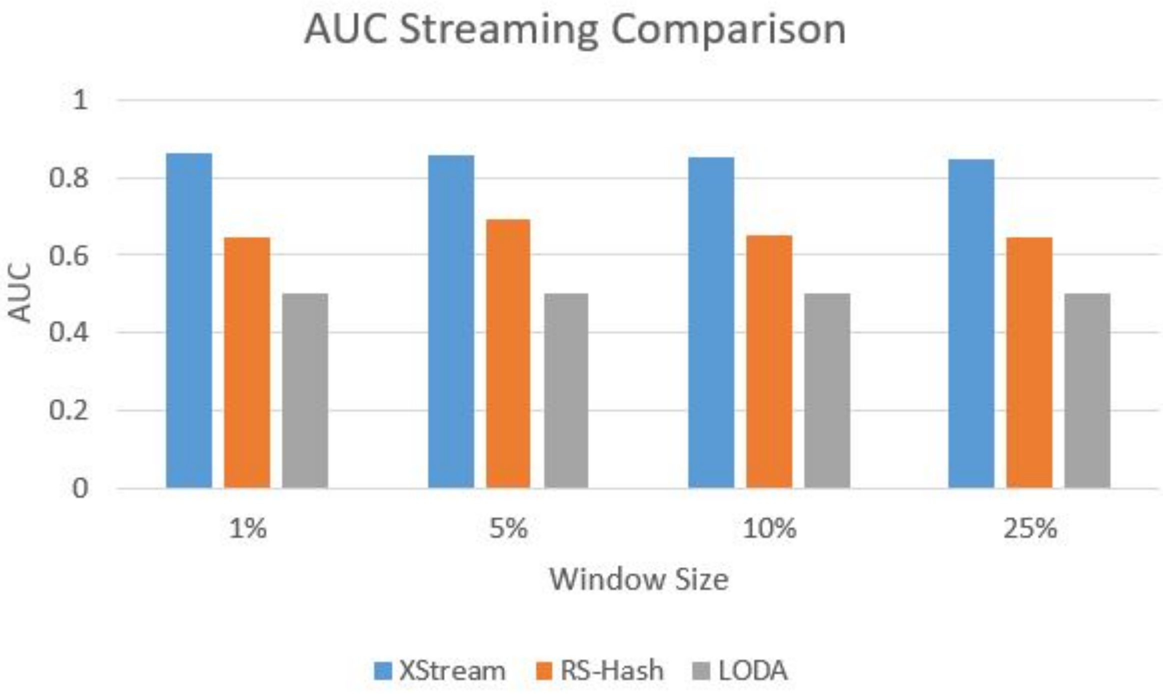


RS-Hash Experiments – OAP Streaming



AUC Comparison

Metric: Area Under ROC Curve (AUC)



Demo using Docker Image

```
docker build -t data_science_project .

docker run -t -i data_science_project
```