Edge Outlier Detection

Huan Li

New Challenges of Anomaly/Outlier Detection

- Unbounded streaming
  - Continuous monitoring, data properties changes over time
- Distributed computing nodes
  - Data isolation and dynamic connectivity
- High-dimensional
  - Multiple built-in sensors reporting in parallel
  - Dim_number > 10, spatial indexing fails
- Paper Digest: Recent Papers on Anomaly Detection – Paper Digest

Contributions

- A general edge-based framework for outlier detection
  - Targeting outlier querying over streaming, distributed, high-dim data
- A set of optimization techniques specific to (p,r)-outliers
  - VLDB’98 definition: (p,r)-outliers are those points who have less than p neighbors laying within r
- A set of optimization techniques specific to LOF-outliers
  - Outliers are those whose density is clearly less than its neighbors’ (a threshold introduced)
  - Maybe as an extension of this work

Outlier Detection Approaches

- Learning-based?
  - Supervised: limited availability of data labels
  - Unsupervised: specific to application scenarios; deployability
  - Data properties changes over a sliding window

- Proximity/Neighborhood-based (all relying on pairwise *distance*!!!)
  - k-distance (k-th shortest distance to neighbors)
  - sum/avg of distances of its kNN
  - number of neighbors within a radius (a density measure)
  - local outlier factor (one's own density compared to neighbors' density)
  - Pros: Easy-to-understand and configurable, generalized to statistical tests, able to use pruning rules

Traditional **Streaming** Outlier Detection

- Window-based (mostly sliding time window)
- Consider insertion and deletion of points
  - Counting and safe pruning for (p,r)-outliers
    - STORM, ABSTRACT-C, MCOD, DUE, Thresh LEAP (VLDB'16), NETS (VLDB'19)
  - Summary of partial points for LOF-outliers – heavier in storage and computation: ILOF, MILOF, DILOF, TADILOF (no code)
- No distributed (no comparison to relevant data from externals)
- Generality: Run locally as a base detector at each device! Even using heterogeneous base detectors!!

Traditional **Distributed** Outlier Detection

Work	Base outlier detector	Setting	remarks
Bhaduri et al.~\cite{bhaduri2011algorithms}	ORCA (based on k-distance)	Distributed network of ring topology	top-N pruning; indexing; randomization
Angiulli et al.~\cite{angiulli2012distributed}	SolvingSet (based on sum of distances of kNN	A supervisor connected to many local n	top-N pruning; incremental communication
Bai et al.~\cite{bai2016efficient}	LOF	A supervisor connected to many local n	grid-based partitioning; spatial indexing
Yan et al.~\cite{yan2017distributed,yan2017distributed	LOF	Hadoop MapReduce	top-N pruning; grid-based partitioning; duplication reduction
Tsou et al.~\cite{tsou2018robust}	One-class Random Forest (normal samples)	Wireless sensor network	weighted ensemble

- One supervisor (central) + multiple workers (local)
- Full dataset originally at central node and is assigned to local nodes
- Local nodes share part of their relevant data and prune inliers
- Final outliers are found at central node

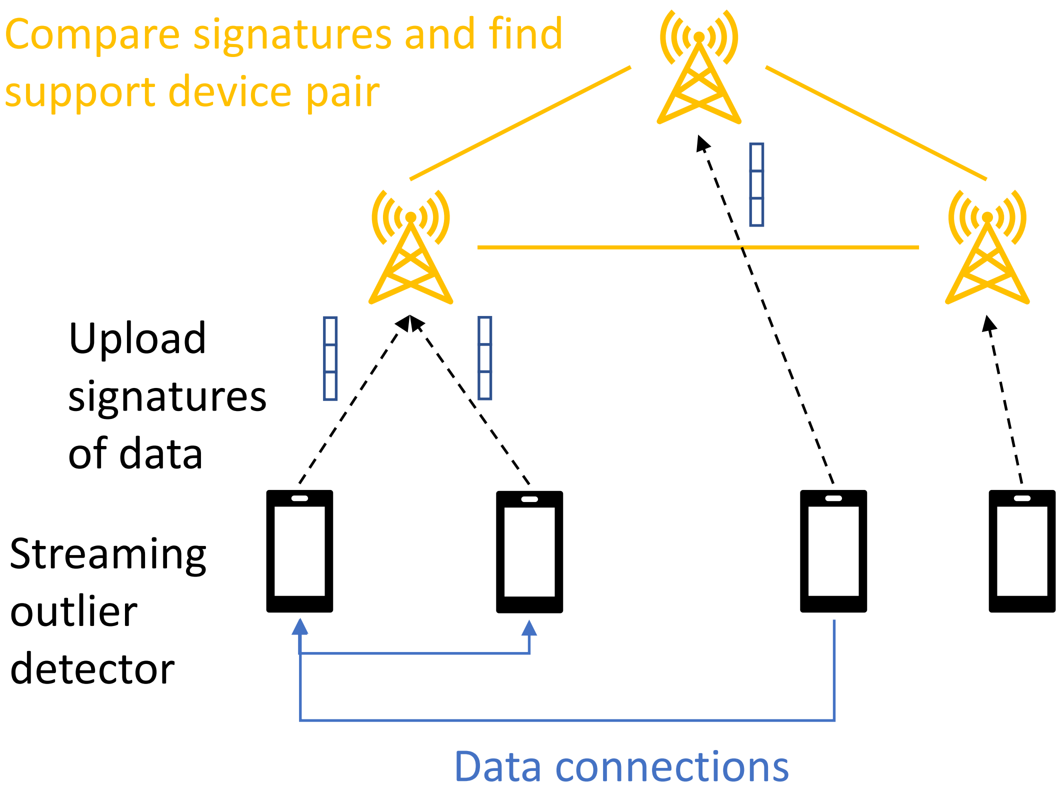
- Dataset not unbounded (multiple passes on secondary storage)
- Set of local nodes (even topology) fixed, non-flexible

Novel Edge Computing Setting

- Devices are collecting high-dimensional data themselves
  - However, outliers are computed by also referring to data from other devices
- Communication between edge node(s) and devices
  - Edge-edge; edge-device; device-device
- Safer to NOT transfer concrete data to edge nodes (decentralized)
- Safer and more cost-effective to transfer LESS data between devices

Our Framework

- Edge node plays the coordinator role
  - Find data nodes (devices) that have neighbors to each other (called a support device)
  - Tell them to connect/disconnect
- Each device runs a streaming outlier detector
  - Get internal data and external data via secure tunnels
  - Independent to finding support devices
- Problem: how to find support devices given large amount of unbounded data points?

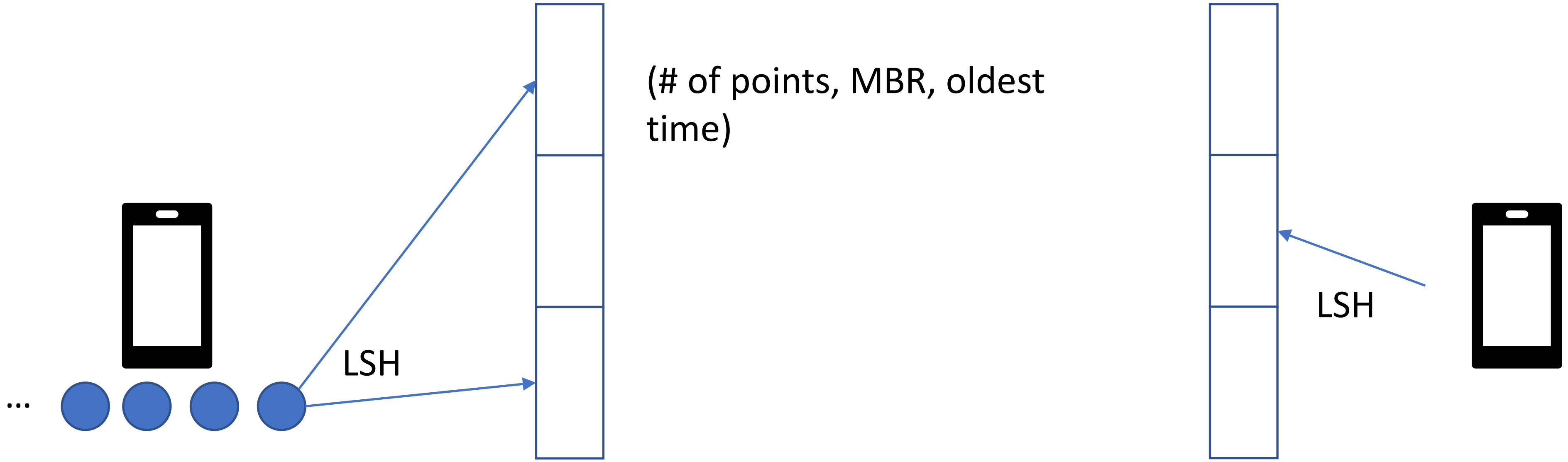


Strategies

- Projection (LSH and the like) --- mapping close high-dim points to the same bin and prune non-support devices quickly
- Data partitioning --- reduce the search space and data transmission
- Bound and pruning (specific to outlier definition)
- Indexing (space-based, reference point-based)
- Data summary (for LOF-outliers)
- Approximation by ignoring insignificant support devices
  - Very small number of relevant data points (estimated by data speed)
  - Too far away and data transmission is of high cost

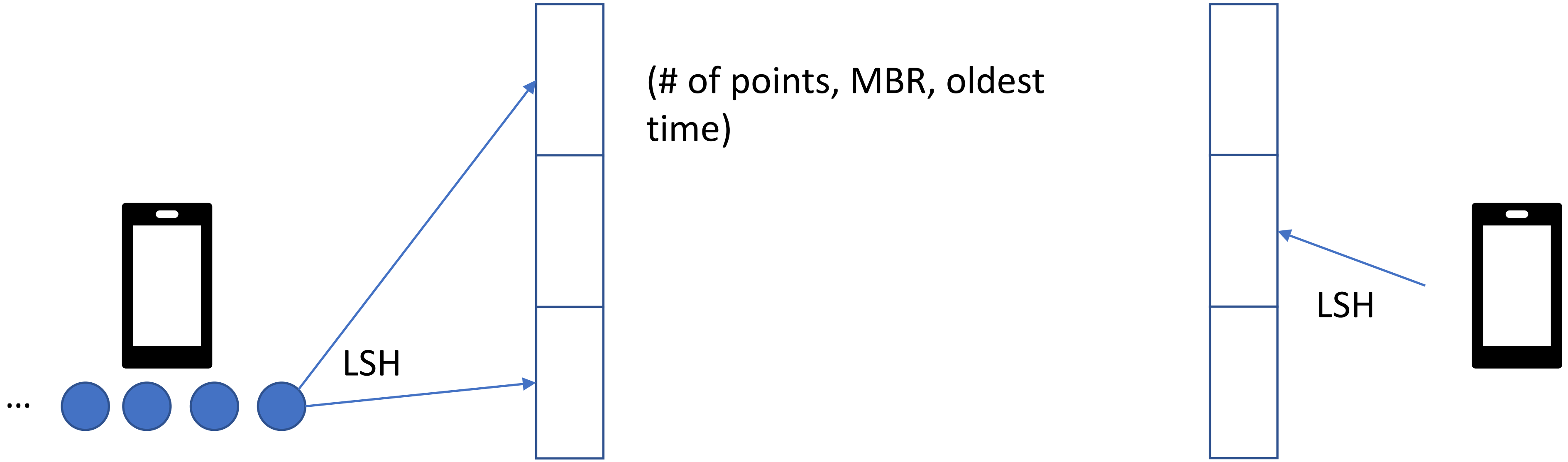
Strategy 1: Projection

The same E2LSH family to all devices



- Two devices’ bins with the same No. are both active, they support each other
- If no such bin pair found, the two devices should not be connected
- Sync up periodically (more frequent, more sensitive to switch connections)
- A signature for a unit interval, incremental processing in edge node

Strategy 2&3: Partitioning and Pruning



- Transfer data at the bin-level instead of device-level
- Edge node level aggregation and pruning
- MBR bounding of a bin – (p,r)-outliers
- Some bins are totally local! -> tighten their concrete outlier scores and use them to prune unpromising active bins (avoid data transfer)

Strategy 4: Approximation

- Observation: an outlier is more likely to be projected onto a bin with fewer points

- A way to model the error due to missing of relevant points
- If a matched bin only contains very few points, ignore it
- Greedy connections: consider nearby devices first, until an error bound is satisfied

Experiments

- Dataset
  - IoT data generator; publicly distributed datasets (ExtraSensory, UCIGas)

- Metrics
  - Effectiveness (precision and recall --- easy to compute ground-truth)
  - Latency (the average/maximum of latencies of devices)
  - Data transmission volume (related to energy)

- Baselines:
  - All data to a central node that runs a streaming outlier detector and returns results to devices
  - Peer-to-peer connection (all-connection or ruled-connection)
  - Ablation study of each strategy

Materials

- Tutorials and blogs
  - yzhao062/anomaly-detection-resources: Anomaly detection related books, papers, videos, and toolboxes (github.com)
  - [CSUR21] 异常检测方法、模型和分类 (I) - 总览
- Datasets
  - UCI Machine Learning Repository: Gas sensors for home activity monitoring Data Set
  - The ExtraSensory Dataset (ucsd.edu)
  - http://odds.cs.stonybrook.edu/
  - 25 Datasets for Deep Learning in IoT | Packt Hub (packtpub.com)
- Implementations
  - JSAT/E2LSH.java at master · EdwardRaff/JSAT (github.com)
  - kykamath/streaming_lsh: A project for clustering text streams using locality-sensitive hashing (LSH) in Python (github.com)
  - Index of /Luan/Outlier/ (usc.edu) VLDB'19
  - kaist-dmlab/NETS: NETS:Extremely Fast Outlier Detection from a Data Stream via Set-Based Processing (github.com) NETS
  - BIT-SYS/CB-ILOF: Cube Based Incremental LOF Algorithm (github.com)

Milestones (tentative)

- Experiments
  - Dataset collection (done)
  - Streaming outlier detectors (done)
  - Suitable LSH implementation - 0708
  - gRPC framework integration - 0722
  - Implementation of strategies of edge nodes – 0805 or 0812
- Paper Writing
  - Definition and Related (partly done)
  - Techniques
  - Experiments
  - Submission – 0901 (PVLDB)