Continuous Machine Learning over Streaming Data

Roger Barga, Nina Mishra, Sudipto Guha, Ryan Nienhuis
Amazon Web Service
Kinesis Streaming Services

Robust Random Cut Forrest

Summary of a dynamic data stream, highly efficient, wide number of use cases...
#Real-time
All data originates in real-time!
But, analytics to gain insights is usually done much, much later.
#WhyWait
Insights are perishable.
Batch analytics operations take too long
Compress the analytics lifecycle
Maximize the value of data

![Diagram](image_url)

- **Business Value**
- **Time To Action**
- **Positive**
- **Negative**

Maximum Business Value
Streaming technology is necessary to detect and act on real-time perishable insights.
#Streaming
Kinesis Data Streaming Services
Get actionable insights quickly

Streaming

Ingest data as it’s generated

Process data on the fly

Real-time analytics/ML, alerts, actions
Streaming with Amazon Kinesis

Easily collect, process, and analyze data and video streams in real time

Kinesis Data Streams
Capture, process, and store data streams

Kinesis Data Firehose
Load data streams into AWS data stores

Kinesis Data Analytics
Analyze data streams with SQL

Kinesis Video Streams
Capture, process, and store video streams
Customer Examples

- **NETFLIX**: Analyze billions of network flows in real-time
- **COMCAST**: Migrated data bus from Kafka to Kinesis
- **SONOS**: 1 billion events per week from connected devices
- **Zillow**: Near-real-time home valuation (Zestimates)
- **THOMSON REUTERS**: Live clickstream dashboards refreshed under 10s
- **GE**: IoT predictive analytics
- **HEARST corporation**: 100 GB/day clickstreams from 250+ sites
- **AdRoll**: 50 billion daily ad impressions, sub-50 ms responses
- **NORDSTROM**: Online stylist processing 10 million events/day
- **lyft**: Facilitate communications between 100+ microservices
Amazon Kinesis
Foundational Service Used Across Amazon

AWS metering  Amazon S3 events  Amazon CloudWatch logs  Amazon.com online catalog  Amazon Go video analytics
Discover actionable insights in real-time

Anomaly Detection
Random Cut Tree

Recurse: The cutting stops when each point is isolated.
Random Cut Forest

Each tree built on a random sample.
Random Sample of a Stream

Reservoir Sampling [Vitter]
Maintain random sample of 5 points in a stream?

- Keep heads with probability $\frac{5}{6}$
- Discard tails with probability $\frac{1}{6}$
Insert – Case I

Start with the Root
If the point falls inside the bounding box follow the path to the appropriate child
Theorem: Insert generates a tree $T' \sim T(\text{[insert data here]})$
What is an Outlier?

Out, liar!

Your theory is wrong!
Anomaly Score: Displacement

A point is an *anomaly* if its insertion greatly increases the tree size ( = sum of path lengths from root to leaves = description length).

Inlier:
Anomaly Score: Displacement

Outlier
NYC Taxi Ridership

Date aggregated every 30 minutes,
Shingle Size: 48

CREATE OR REPLACE PUMP "STREAM_PUMP" AS
  INSERT INTO "TEMP_STREAM"
  SELECT STREAM "passengers", "distance", ANOMALY_SCORE
  FROM TABLE (RANDOM_CUT_FOREST (
    CURSOR(SELECT STREAM * FROM "SOURCE_SQL_STREAM"))
-- creates a temporary stream.
CREATE OR REPLACE STREAM "TEMP_STREAM" (
  "passengers" INTEGER,
  "distance" DOUBLE,
  "ANOMALY_SCORE" DOUBLE);
-- creates another stream for application output.
CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" (
  "passengers" INTEGER,
  "distance" DOUBLE,
  "ANOMALY_SCORE" DOUBLE);
-- Compute an anomaly score for each record in the input stream
-- using Random Cut Forest
CREATE OR REPLACE PUMP "STREAM_PUMP" AS
  INSERT INTO "TEMP_STREAM"
  SELECT STREAM "passengers", "distance", ANOMALY_SCORE
  FROM TABLE (RANDOM_CUT_FOREST (CURSOR(SELECT STREAM * FROM "SOURCE_SQL_STREAM")))
-- Sort records by descending anomaly score, insert into output stream
CREATE OR REPLACE PUMP "OUTPUT_PUMP" AS
  INSERT INTO "DESTINATION_SQL_STREAM"
  SELECT STREAM * FROM "TEMP_STREAM"
  ORDER BY FLOOR("TEMP_STREAM".ROWTIME TO SECOND), ANOMALY_SCORE DESC;
NYC Taxi Data

numPassengers

0 5000 10000 15000 20000 25000 30000 35000 40000 45000
2014-09-16 21:30:00 2014-09-19 13:00:00 2014-09-22 04:30:00 2014-09-24 20:00:00 2014-09-27 11:30:00 2014-09-30 03:00:00 2014-10-02 18:30:00 2014-10-05 10:00:00 2014-10-08 01:30:00 2014-10-10 17:00:00 2014-10-13 08:30:00 2014-10-16 00:00:00 2014-10-18 15:30:00 2014-10-21 07:00:00 2014-10-23 22:30:00 2014-10-26 14:00:00 2014-10-29 05:30:00 2014-10-31 21:00:00 2014-11-03 12:30:00 2014-11-06 04:00:00 2014-11-08 19:30:00 2014-11-11 11:00:00 2014-11-14 02:30:00 2014-11-16 18:00:00 2014-11-19 09:30:00 2014-11-22 01:00:00 2014-11-24 16:30:00 2014-11-27 08:00:00 2014-11-30 20:00:00 2014-12-02 15:00:00 2014-12-05 06:30:00 2014-12-07 22:00:00 2014-12-10 13:30:00 2014-12-13 05:00:00 2014-12-15 20:30:00 2014-12-18 12:00:00 2014-12-21 03:30:00 2014-12-23 19:00:00 2014-12-26 10:30:00 2014-12-29 02:00:00 2015-01-03 09:00:00 2015-01-06 00:30:00 2015-01-08 16:00:00 2015-01-11 07:30:00 2015-01-13 23:00:00 2015-01-16 14:30:00 2015-01-19 06:00:00 2015-01-21 21:30:00 2015-01-24 13:00:00 2015-01-27 04:30:00 2015-01-30 17:30:00
Robust Random Cut Forest Based Anomaly Detection On Streams

Sudipto Guha  
University of Pennsylvania, Philadelphia, PA 19104.

Nina Mishra  
Amazon, Palo Alto, CA 94303.

Gourav Roy  
Amazon, Bangalore, India 560055.

Okke Schrijvers  
Stanford University, Palo Alto, CA 94305.

SUDIPTO@CIS.UPENN.EDU  
NMISHRA@AMAZON.COM  
GOURAVR@AMAZON.COM  
OKKES@CS.STANFORD.EDU

Abstract

In this paper we focus on the anomaly detection problem for dynamic data streams through the lens of random cut forests. We investigate a robust random cut data structure that can be used as a sketch or synopsis of the input stream. We provide a plausible definition of non-parametric anomalies based on the influence of an unseen point on the remainder of the data, i.e., the externality imposed by that point. We show how the sketch can be efficiently updated in a dynamic data stream. We demonstrate the viability of the algorithm on publicly available real data.

a point is data dependent and corresponds to the externality imposed by the point in explaining the remainder of the data. We extend this notion of externality to handle "outlier masking" that often arises from duplicates and near duplicate records. Note that the notion of model complexity has to be amenable to efficient computation in dynamic data streams. This relates question (1) to question (2) which we discuss in greater detail next. However it is worth noting that anomaly detection is not well understood even in the simpler context of static batch processing and (2) remains relevant in the batch setting as well.

For question (2), we explore a randomized approach, akin to (Liu et al., 2012), due in part to the practical success reported in (Emmott et al., 2013). Randomization is a powerful tool and known to be valuable in supervised learn-
Attribution and Directionality

Explainable/Transparent/Interpretable ML

“If my time-series data with 30 features yields an unusually high anomaly score. How do I explain why this particular point in the time-series is unusual? [..] Ideally I’m looking for some way to visualize “feature importance” for a specific data point.”

--- Robin Meehan, Inasight.com
What is Attribution?

It’s the ratio of the “distance” of the anomaly from normal. (It’s a distance in space of repeated patterns in the data.)

\[ \Delta^i(p) = \frac{(\text{Score}^{+i}(p) - \text{Score}^{-i}(p))}{\text{Score}^{+i}(p)} \]
What is Attribution?

Anomaly Score
## NYC Taxi Ridership Data

<table>
<thead>
<tr>
<th>Pickup Time</th>
<th>Dropoff Time</th>
<th>Distance</th>
<th>Base Fare</th>
<th>Surcharge</th>
<th>Tax</th>
<th>Tip</th>
<th>Tolls</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>7/1/14 1:43</td>
<td>7/1/14 1:51</td>
<td>2.59</td>
<td>9.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>10.5</td>
</tr>
<tr>
<td>7/1/14 1:33</td>
<td>7/1/14 1:47</td>
<td>2.38</td>
<td>12</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>7/1/14 1:37</td>
<td>7/1/14 1:50</td>
<td>2.87</td>
<td>11.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>12.5</td>
</tr>
<tr>
<td>7/1/14 1:35</td>
<td>7/1/14 1:50</td>
<td>4.68</td>
<td>16</td>
<td>0.5</td>
<td>0.5</td>
<td>4.95</td>
<td>0</td>
<td>21.95</td>
</tr>
<tr>
<td>7/1/14 1:25</td>
<td>7/1/14 1:49</td>
<td>6.72</td>
<td>23</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>7/1/14 1:30</td>
<td>7/1/14 1:50</td>
<td>5.04</td>
<td>18.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>19.5</td>
</tr>
<tr>
<td>7/1/14 0:17</td>
<td>7/1/14 0:24</td>
<td>2.53</td>
<td>9</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>7/1/14 0:06</td>
<td>7/1/14 0:22</td>
<td>2.48</td>
<td>12.5</td>
<td>0.5</td>
<td>0.5</td>
<td>2.6</td>
<td>0</td>
<td>16.1</td>
</tr>
<tr>
<td>7/1/14 0:17</td>
<td>7/1/14 0:24</td>
<td>1.81</td>
<td>7.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>8.5</td>
</tr>
<tr>
<td>7/1/14 1:38</td>
<td>7/1/14 1:50</td>
<td>6.26</td>
<td>19</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td>21</td>
</tr>
</tbody>
</table>

---

The Moving Example

A Fan/Turbine

1000 pts in each blade
Gaussian, for simplicity
Blades designed unequal
Rotate counterclockwise
3 special “query” points
100 trees, 256 points each

$p1=(-0.4,0)$
$p2=(0,-0.4)$
$p3=(0.4,0)$
Anomaly Score at P1

Blade overhead = Not an anomaly

What is going on at 90 degrees?
All 3 Blades

Anomaly Score at p1
Anomaly Score at p2
Anomaly Score at p3

Degree of Rotation
**Transparent Attributions**

x coordinate’s contribution for p1?

p1 is far away in x-coord most of the time

But what is happening to y?
Directionality

Sharp transition when the blade moves from above to below at p1!
Total score plummets.

Slowly rotating away
Total score remains high

Blade is below
Blade is above
Anomaly Score at p1 Contribution y (sum)
Hotspots on a Stream
Hotspots on a Stream
Detecting Anomalies in Directed Graphs

Targeted change

Non-targeted change

Suspiciousness
Time Series Forecasting
False Alarms

Anomaly Detection with user feedback

**Alarm fatigue:** Personnel become desensitized

560 alarm related deaths during 2005—2008 (FDA data)

“Alarms sounded 1 hour before the nurse discovered he was unresponsive. He eventually died. An investigation found the alarm volume had been turned off.”
System View

Streaming semi-supervised learning

Data

Is it an anomaly?

Anomaly Detection

Is it interesting?

Semi-Supervised Classification

Alarm / No alarm

\(\sqrt{\text{RRCF}}\)
Orders Data: vs.

Orders per minute

RRCF anomalies

Ground truth labels

Alarms

Labeled examples
Robust Random Cut Forrest

Summary of a dynamic data stream, efficient, number of use cases...

- Anomaly Detection
- Attribution and Directionality
- Hotspot Detection
- Classification
- Forecasting
- Missing Value Imputation
- Anomaly Detection in Streaming Directed Graph
Amazon Kinesis Data Analytics
The easiest way to use machine learning!

Available Now
• Anomaly Detection
• Anomaly Detection with explanations
• Hotspot Detection (releasing soon!)

Coming Soon
• Classification
• Time Series Forecasting
• Missing Value Imputation
Contributors To This Project

Roger Barga, Kapil Chhabra, Charles Elkan, Dhivya Eswaran, Christos Faloutsos, Praveen Gattu, Gaurav Ghare, Sudipto Guha, Shiva Kasiviswanathan, Nina Mishra, Morteza Monemizadeh, Lauren Moos, Yonatan Naamad, Ryan Nienhuis, Gourav Roy, Okke Schrijvers, Joshua Tokle, and Tal Wagner