DNS Threat Detection

Blueprint X-Challenge Q1’22

Threat Detection

What do end users care about?

End Users Care About Threats

Threats vs. Anomalies

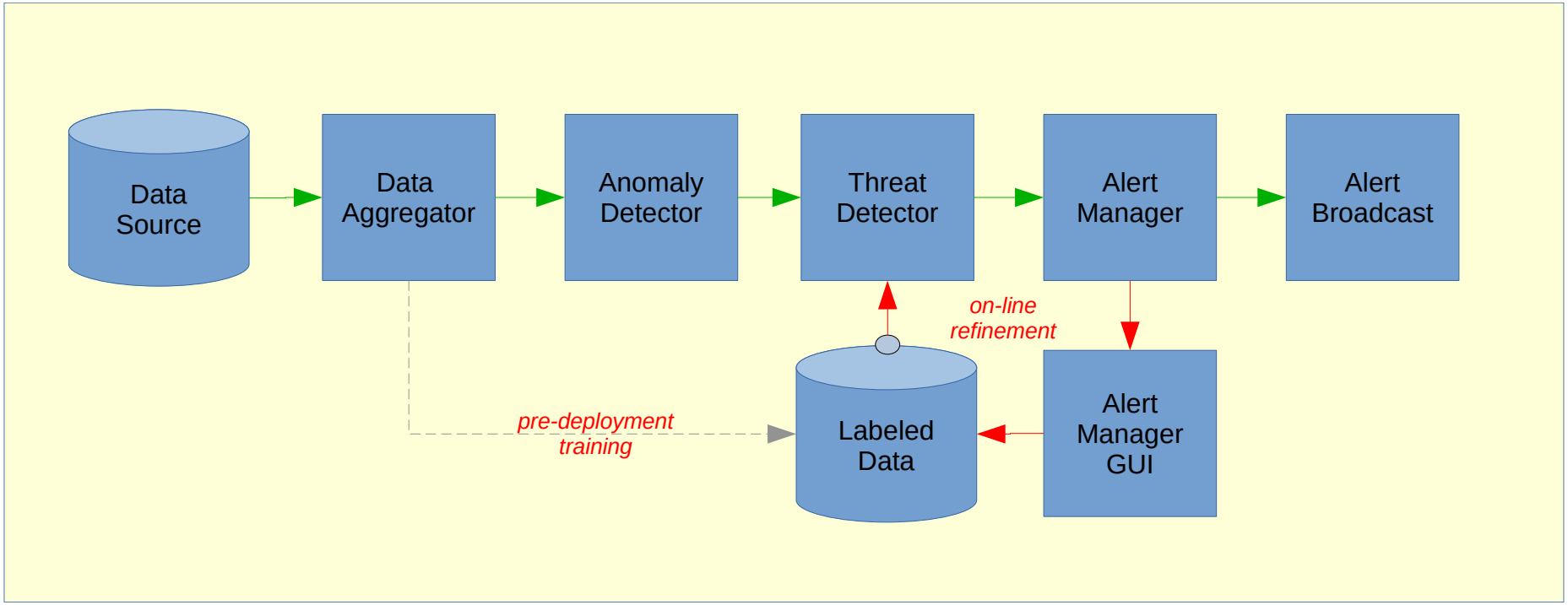
Threats Depend on User Context

Threats are Relevant Anomalies

A Threat Detection System

What makes up a Threat Detection System?

A threat-detection system must recognize anomalies in a data source and selectively report those anomalies that align with the end-user's notion of threat.



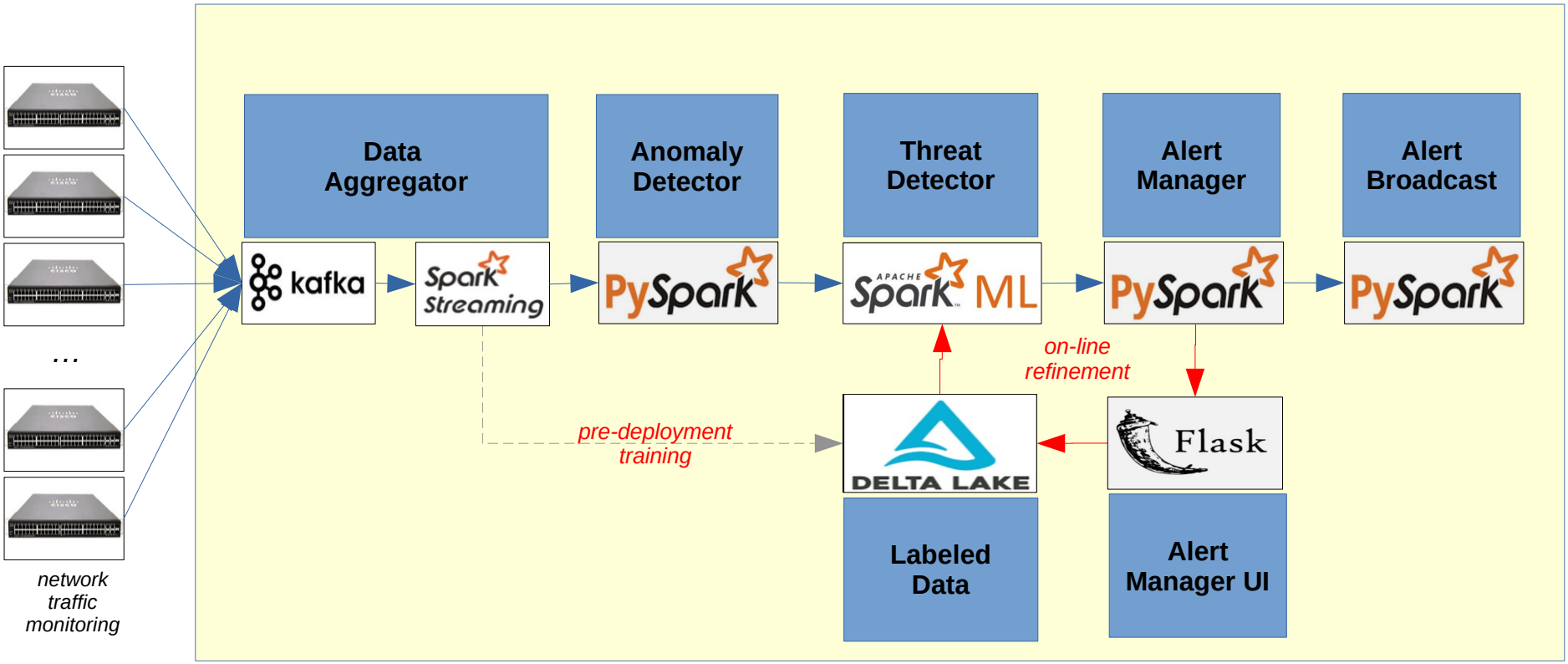
Typical approaches include:

- Pre-deployment training of ML classifiers for desirable alarms using existing threat definitions
- Post-deployment refinement by user contributed threat definitions from a user interface
- Some combination of the two approaches

A Threat Detection System

What might a Network Threat Detection System Look Like?

A threat-detection system must recognize anomalies in a data source and selectively report those anomalies that align with the end-user's notion of threat.

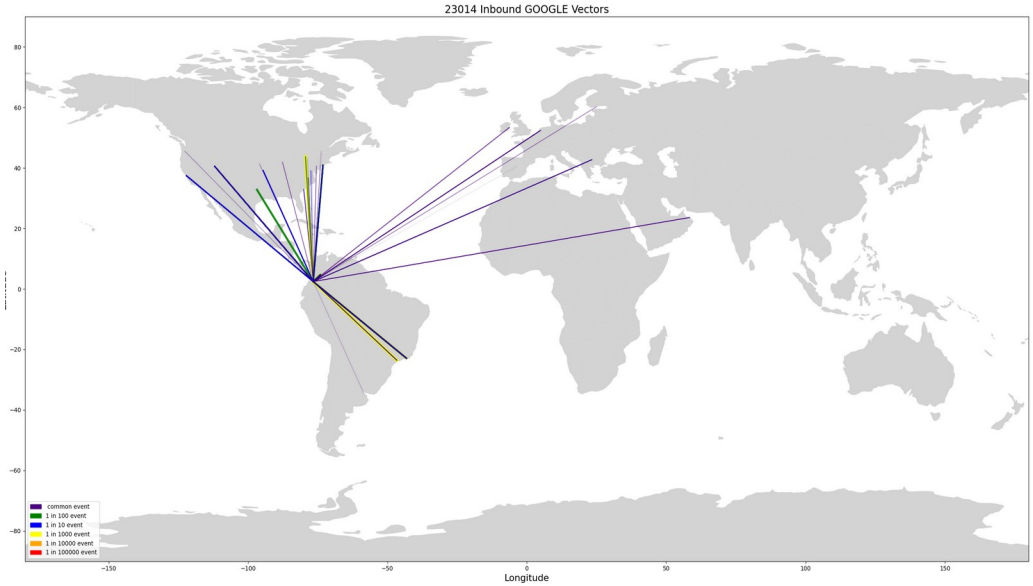


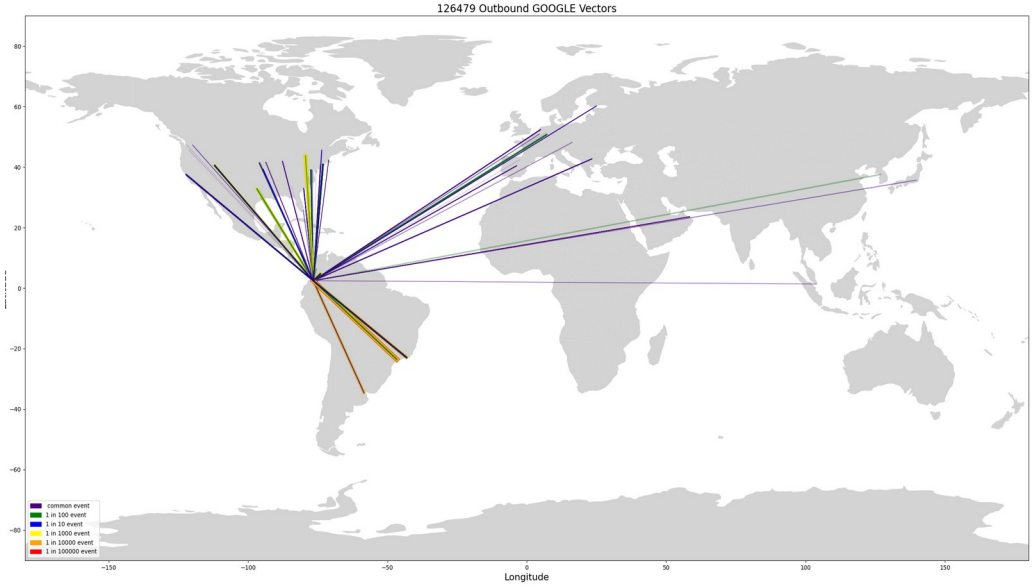
Typical approaches include:

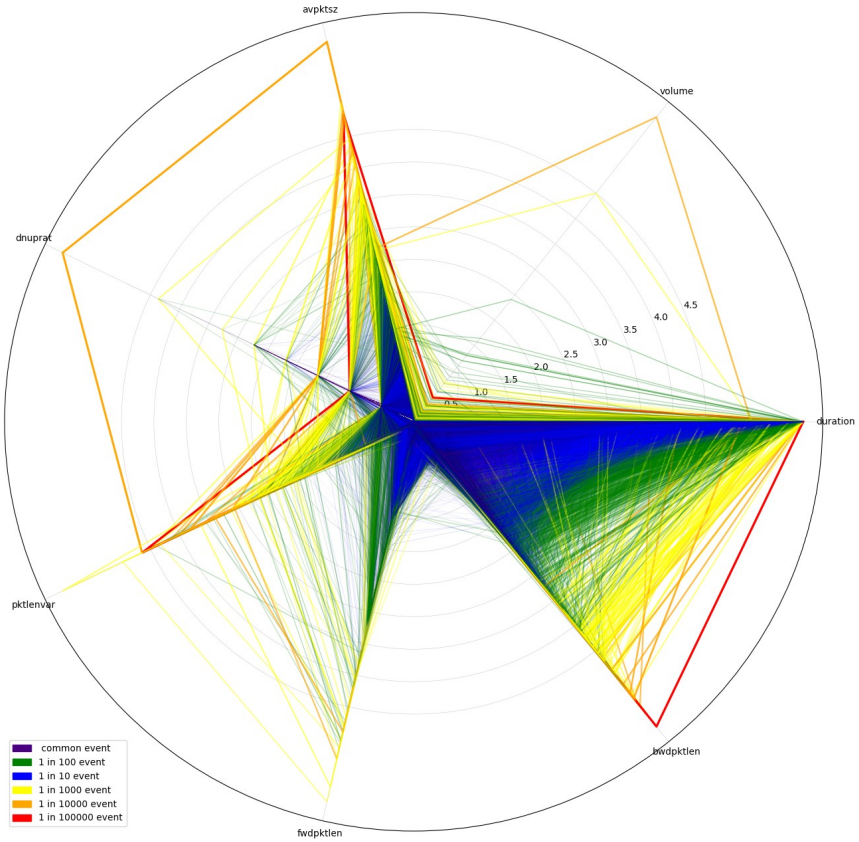
- Pre-deployment training of ML classifiers for desirable alarms using existing threat definitions
- On-line refinement by user contributed threat definitions from a graphical interface
- Some combination of the two approaches

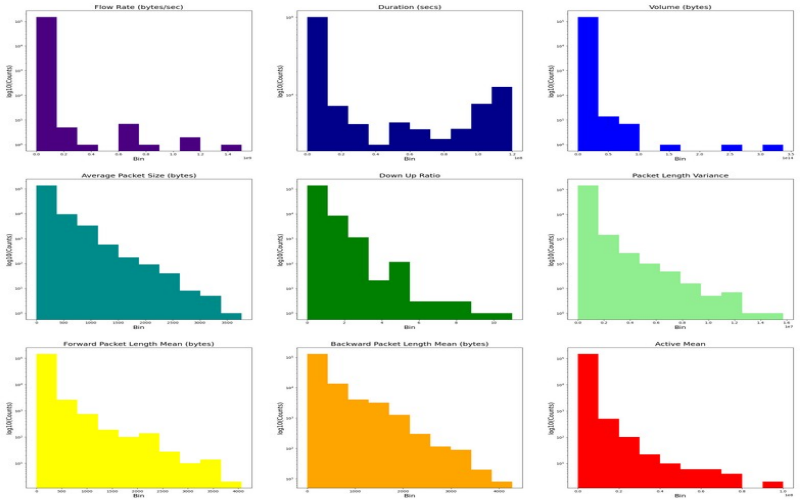
Sample Results #1

GOOGLE Traffic



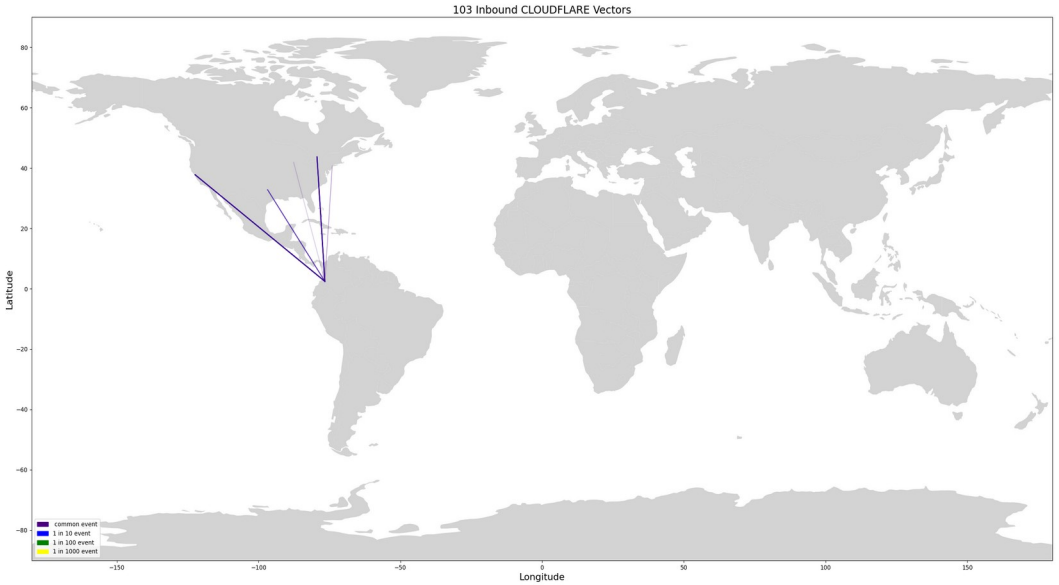


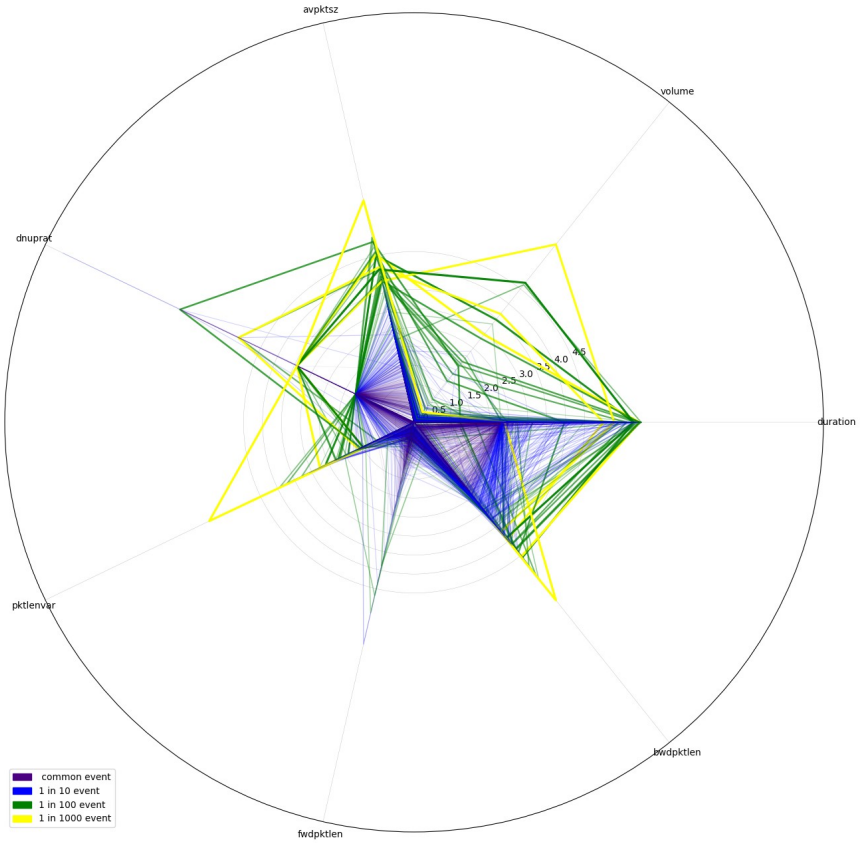


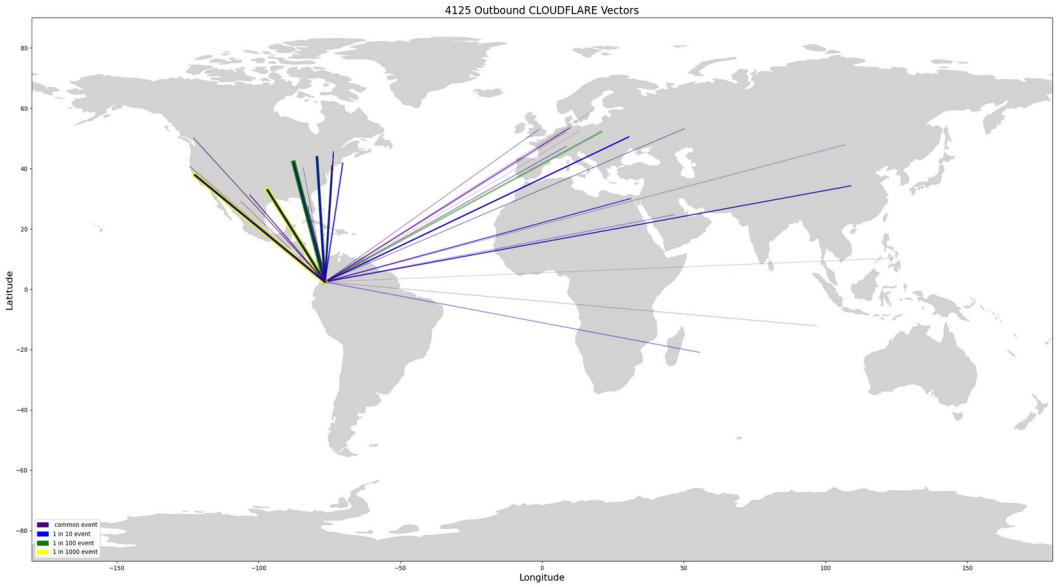


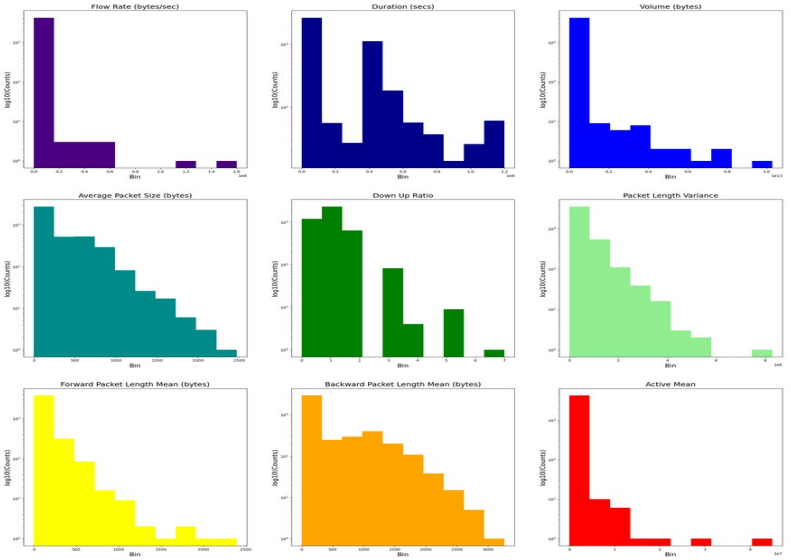
Sample Results #2

CLOUDFLARE Traffic









Discussion

Anomalies, Threats, False Alarms

- Identifying an Anomaly is based on *statistics*
- Identifying a Threat is based on *relevance*
- Data Science techniques excel at recognizing Anomalies, however…
  - *...most Anomalies are **not** Threats*
- Successful Threat Detection Systems avoid irrelevant anomalies as threats

Novel Differentiation

How does this differ from the original accelerator?

Approach:

- Anomaly detection
- Initial Threat definition from historical data (optional)
- Active refinement of Threat Model (online training with user input)
- Alert Manager GUI

Techniques (not yet implemented)

- Kafka & SparkStreaming (to handle massive data volume)
- Pipelined decomposition of traffic into “FlowLabeler” format
- On-line Deep-Learning refinement of model
  - *Model adapts as users suppy inputs*
  - *Model adapts as nature of traffic evolves*
- Web based alerting and assessment UI

Market Alignment

How quickly could this be made into a product for our customers?

Market Alignment:

- Scalability Choices:
  - *Approach 1. Massive Scalability (use DataBricks ecosystem)*
    - Approach 1a. uses python and pyspark with DataBricks
    - Approach 1b. uses Scala and SparkML with DataBricks
  - *Approach 2. Modest Scalability (use standard Data Science ecosystem)*

- Elements Needed
  - *Need to develop the threat detector*
  - *Need to develop the alert manager UI*
  - *Need to fully flesh out the training loop*
  - *Need to implement the streaming pipeline*
  - *Need to port various bits and pieces*

- Development Timescales:
  - *Approach #2 – Standard Data Sci stack is mature*
    - Fast, perhaps 10-12 weeks to MVP
  - *Approach #1a – PySpark + Python Data Sci on DataBricks*
    - Medium, perhaps 15-18 weeks to MVP
  - *Approach #1b – Scala + SparkML + 1st Principles Data Sci on DataBricks*
    - Slow, perhaps 24-28 weeks to MVP

Partnership Alignment

How difficult would it be to adapt this for our partners to use?

Partner Uptake:

- Nearly all of the proposed technologies
  - *Kafka, SparkStreaming, PySpark, SparkML, DeltaLake are found in the DataBricks ecosystem*

Appendices

Useful References

- “Dataset-Unicauca-Version2”
  - From https://www.kaggle.com/datasets/jsrojas/ip-network-traffic-flows-labeled-with-87-apps
  - 3.58M rows of network traffic data with 87 statistical features captured from the Universidad Del Cauca, Popayan, Columbia on April 26th, 27th, 28th, and May 9th, 11th, and 15th, 20173
  - The data was captured using the ‘FlowLabeler’ below

- FlowLabeler
  - See https://github.com/jsrojas/FlowLabeler
  - This is a tool for processing either pcap files or live streaming data and producing formatted data containing bidirectional statistics and the application layer protocol associated. It uses the low-level nDPI library to produce the ‘csv’ files that we’re working with...
    - Requires: pandas, numpy, dpkt, lru-dict, pypcap (note that pypcap requires libpcap-dev)

- “The Mean/Max Statistic in Extreme Value Analysis”
  - Rochet and Serra, https://arxiv.org/abs/1606.08974 (2016-06-19)