Eliminating Barriers to Automated Tensor Analysis for Large-scale Flows

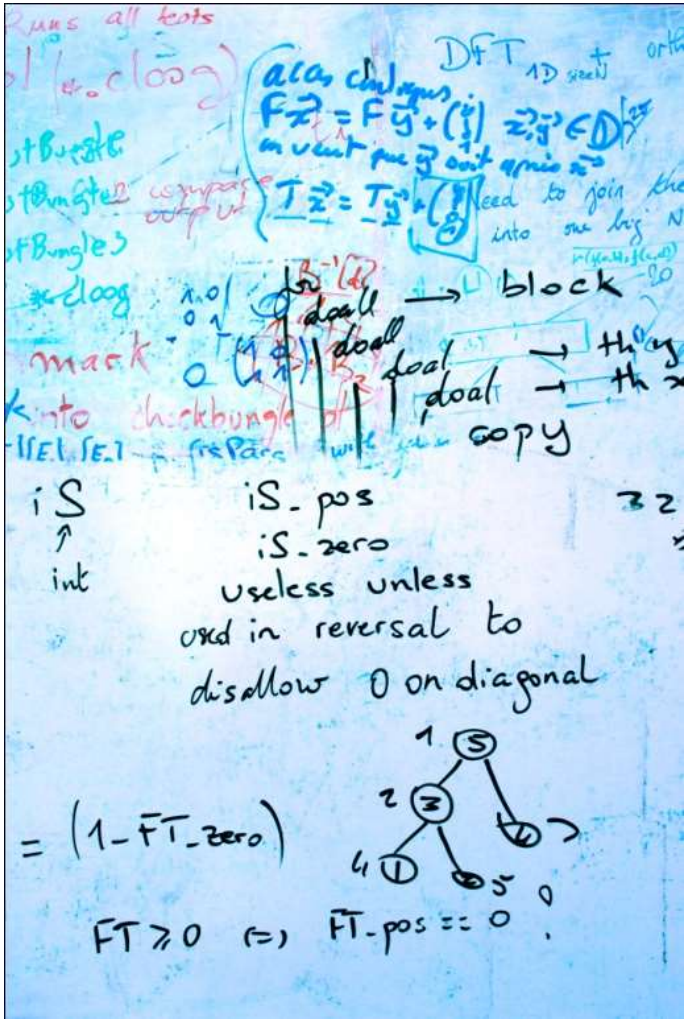
FloCon 2018

An approach to data analysis in support of security operations based on R-Scope® and ENSIGN™

James Ezick
Muthu Baskaran
Alan Commike
Aditya Gudibanda
Thomas Henretty
M. Harper Langston
Pierre-David Letourneau
Jordi Ros-Giralt
Richard Lethin

Reservoir Labs, Inc.
New York, NY

11 January 2018

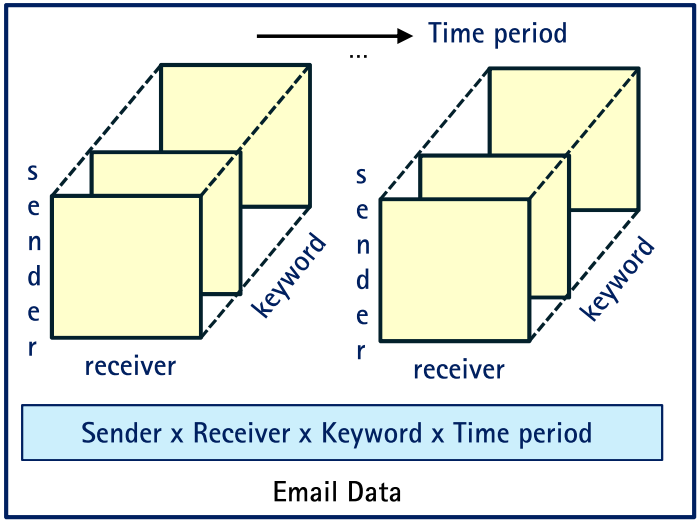


www.reservoir.com

Tensors: Representing Multidimensional Data

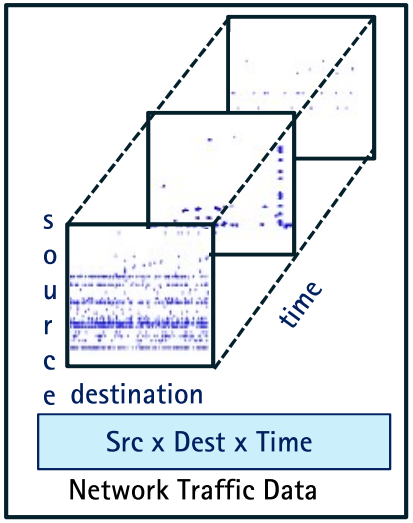
Real World Data

- Multidimensional
- Heterogeneous
- Large
- Sparse



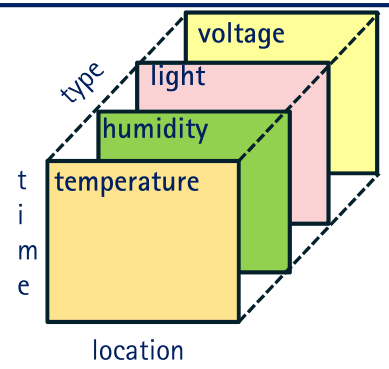
Sender x Receiver x Keyword x Time period

Email Data



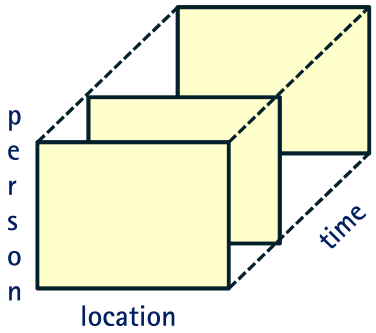
Src x Dest x Time

Network Traffic Data



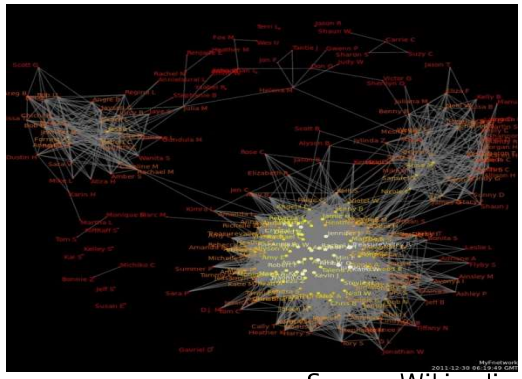
Time x Location x Type

Environmental Sensor Monitoring



Person x Location x Time

Physical Access Data

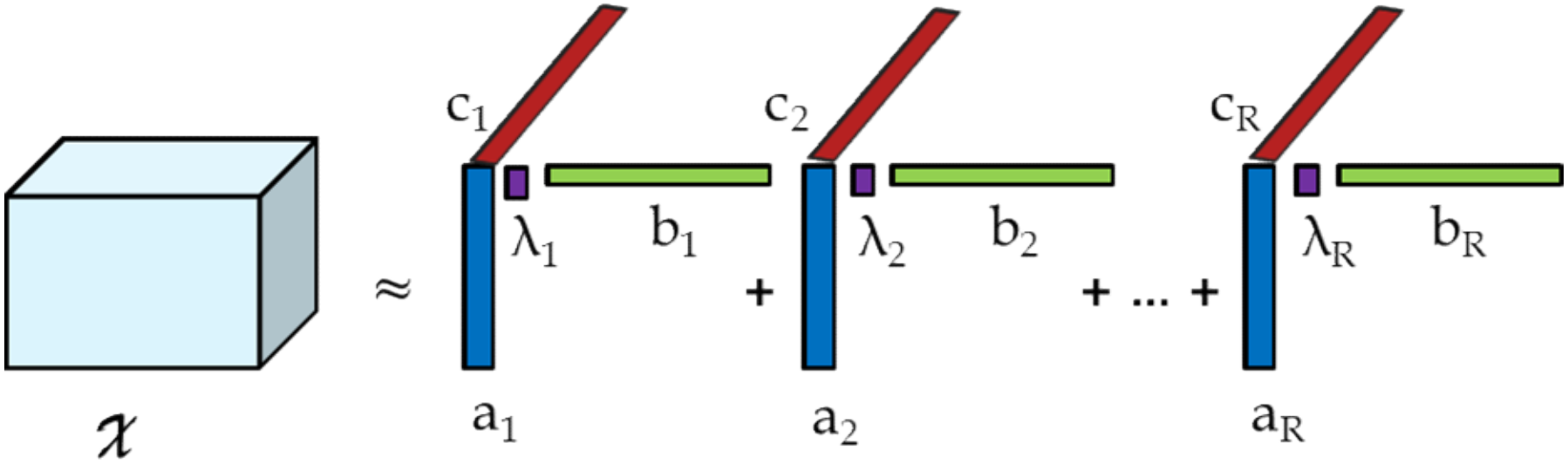


Source: Wikipedia

Person x Person x Relation

Social Network Graph

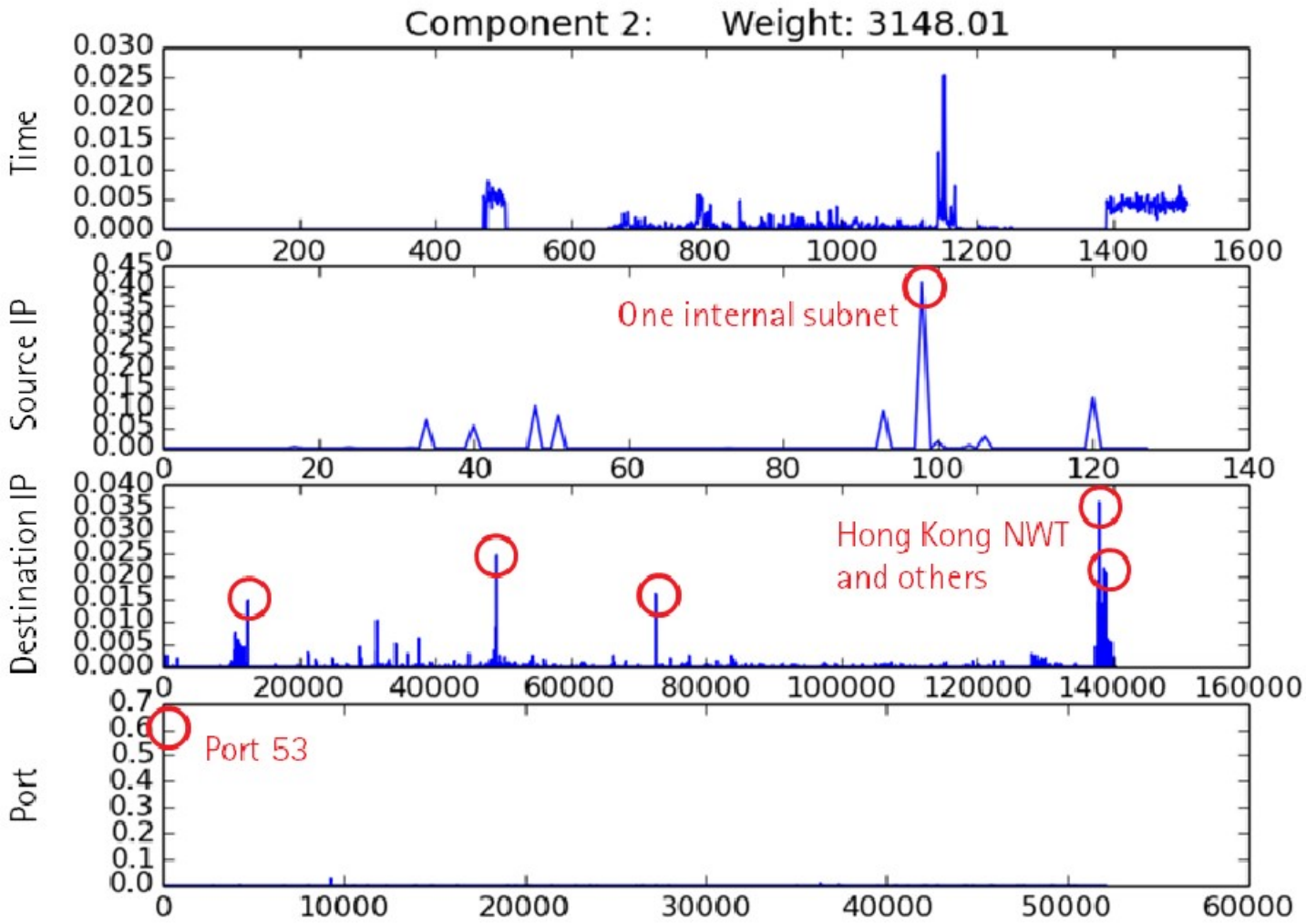
Basic CP Tensor Decomposition



CP Tensor Decomposition

Tensor is decomposed into a non-unique weighted sum of a pre-defined number of rank-1 components

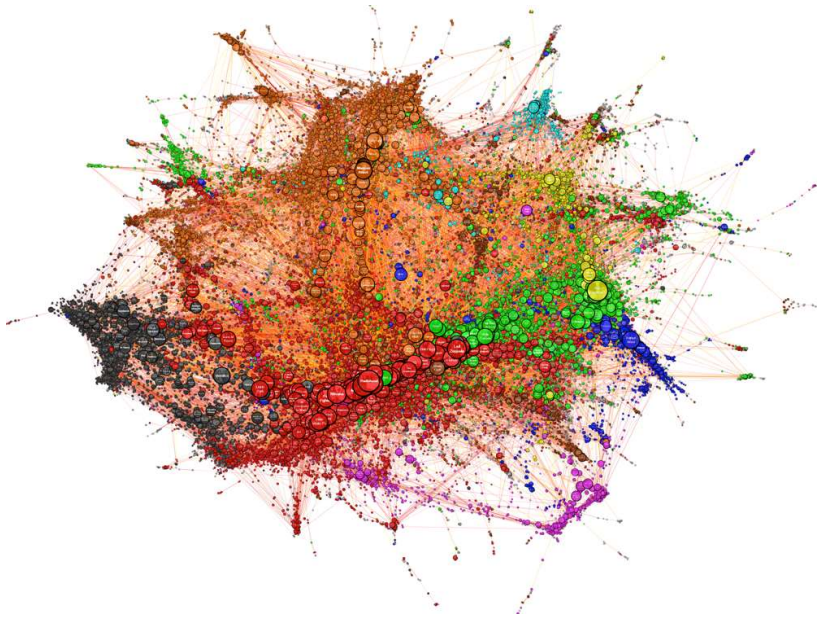
Example: Suspicious DNS Traffic



Time x Source IP x Destination IP x Port

Finding the Patterns First

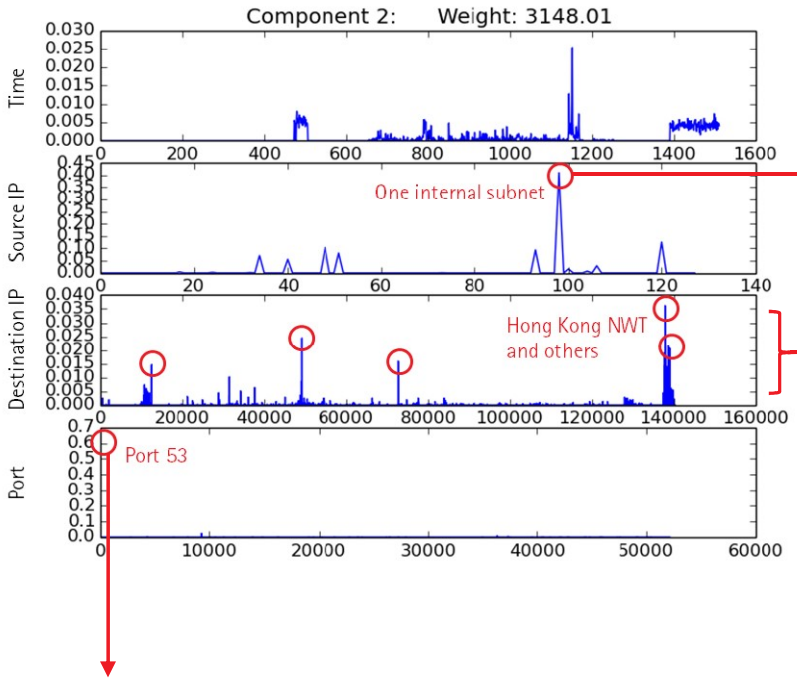
Data Collect



Pattern Discovery

Can we supplement traditional incident-focused approaches to threat discovery with an approach that feeds metadata to a pattern-focused analytic?

Investigation: Suspected Data Exfiltration

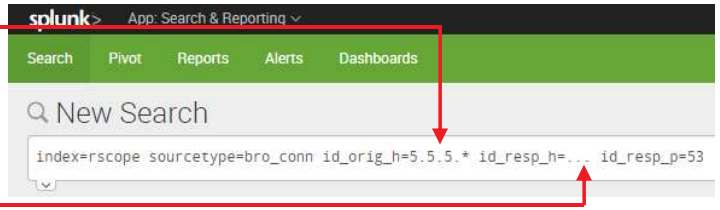


Port 53: Used for DNS

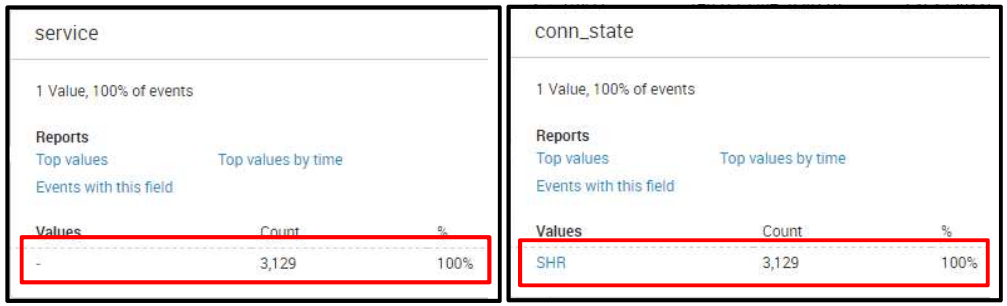
- None of the 4-6 high scoring Destination IPs are the SCinet DNS server
- All destinations are external IPs

Investigating

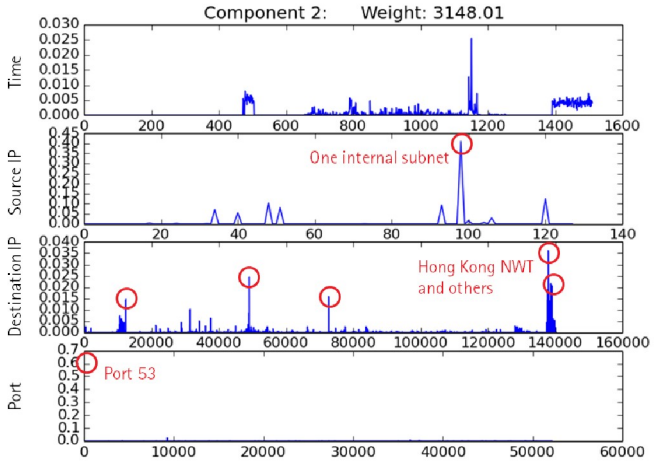
- Search for all connections from this subnet to these destinations with Port 53

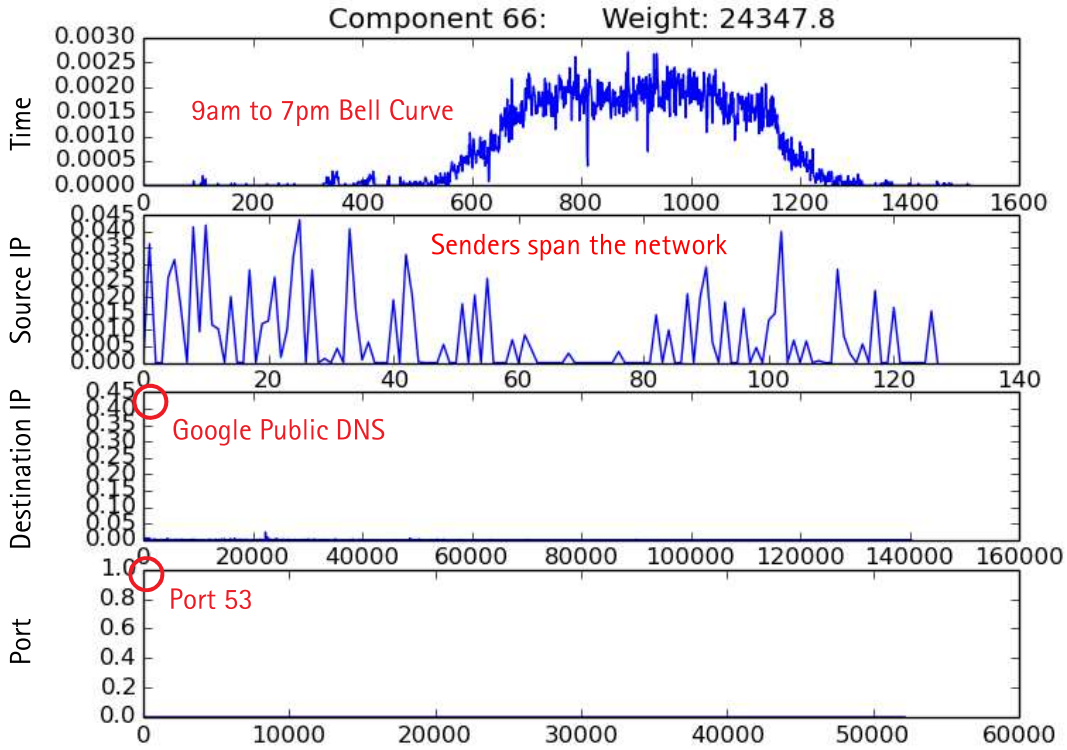


- There is only one IP in this subnet going to those destinations (this is a single internal actor)
- R-Scope labels the service as "-" (not recognized). This is not DNS traffic.
- Connection state is SHR: responder recognized the messages, half-opened communication, and then terminated the connection



Investigation: Suspected Data Exfiltration





Compare to Typical DNS Traffic

- The component on the right shows an example of typical DNS traffic
- The only Destination IPs are Primary (8.8.8.8) and Secondary (8.8.4.4) DNS
- The time dimension is a highly regular curve between 9 am and 7 pm – the running hours of the conference

ENSIGN vs. SCinet: Pattern Discovery at Scale

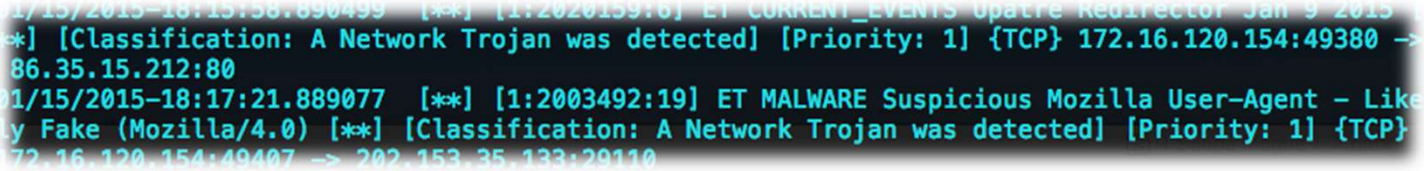
Vital Stats (2015-2017)

- Volunteers
- "Fastest Network in the World"
- ~40 Tap Points of 1G, 10G, 100G, ~200 wireless Aps
- SC: ~12,000 Researchers & Professionals
- 350+ Booths
- No Firewall
- Data Collect from Multiple R-Scope boxes
- No Asset Identification or Authentication
- No Export of Identifying Data

Using ENSIGN …

We have uncovered and visualized patterns indicative of:

- Distributed port scans evolving to machine takeover
- Distributed denial of service attacks
- DNS-based data exfiltration/insider threat
- SSH password guessing (apart from scanning)
- Network policy violations
- Exploitation of application-specific port vulnerabilities
- Patterns of traffic indicative of scans for printers or IoT devices
- Broken or misconfigured network services
- Selective, persistent use of cryptographic methods in point-to-point communication



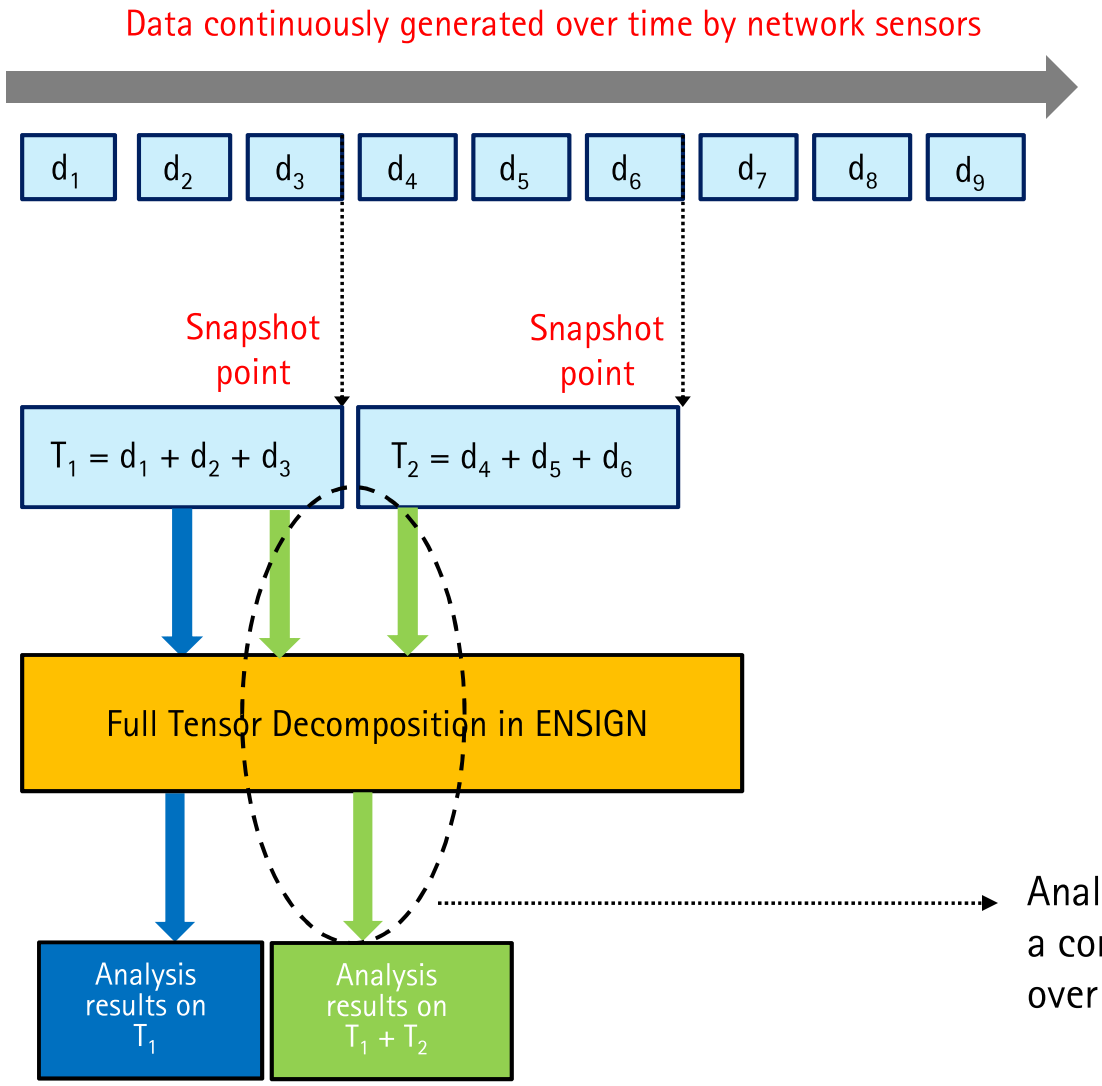
What are the barriers to automated analysis?

1. Provide a faster response to emerging events

2. Automate the classification of components

3. Demonstrate a repeatable process

The Barrier – Respond to Emerging Events

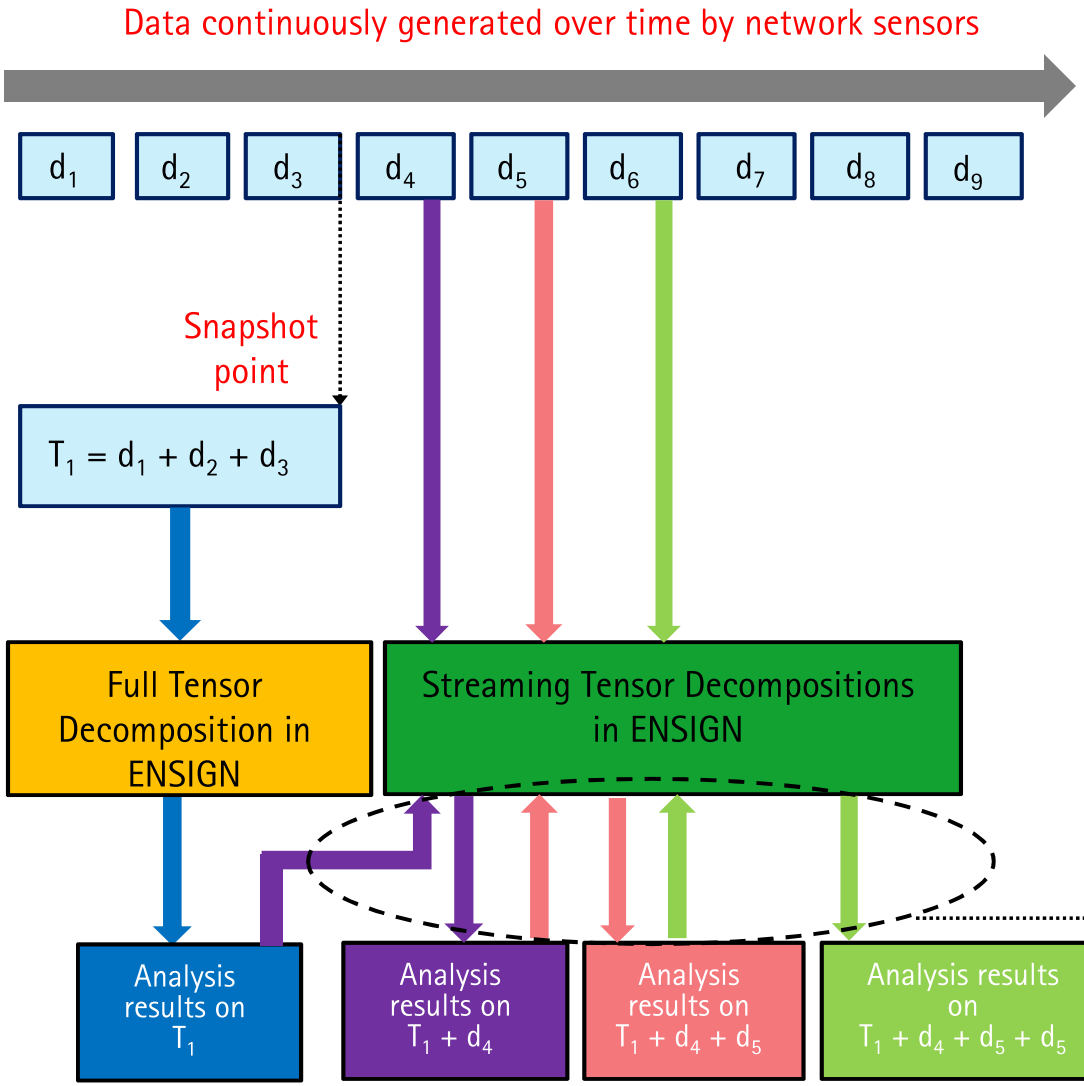


Full Tensor Decomposition

- Complete contextual insights
- Computationally intensive
- **Not ideal for real-time updates**

Analysis of all data is needed to give a complete picture of the activities over the entire timeframe

Streaming Tensor Decompositions

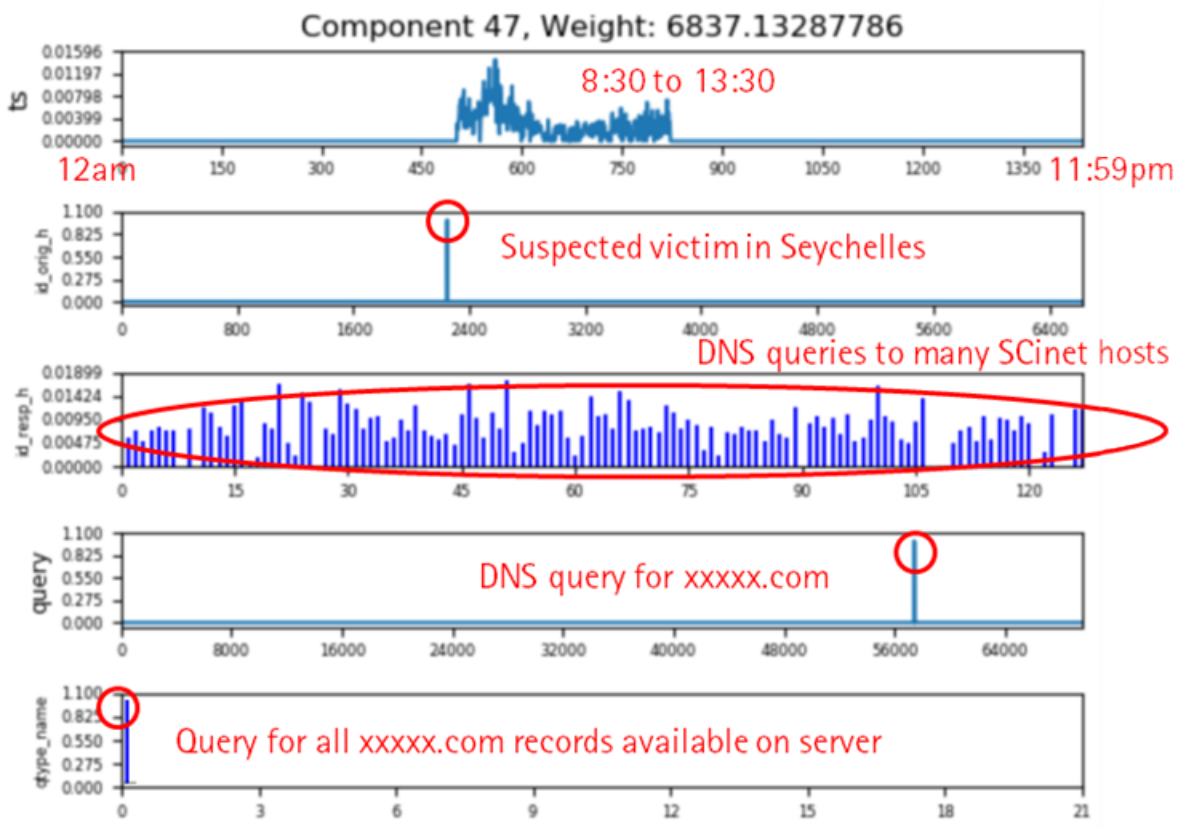


Streaming Tensor Decompositions

- Gives contextual insights with reasonable precision
- Low computational complexity
- Nearer to real-time updates

Previous decomposition and only new data streams are processed and analyzed to give a complete picture

DNS Amplification DDoS Attack



DNS amplification DDoS attack identified from full tensor decomposition:

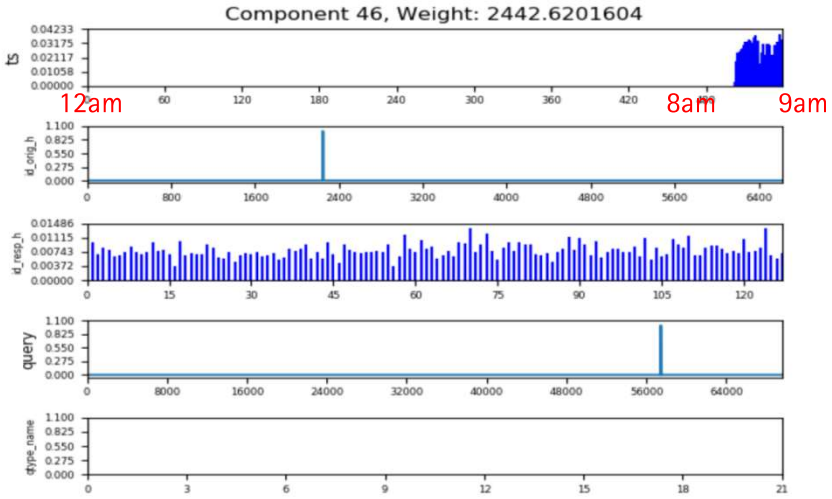
- Analyzed DNS log collected over 24 hours (12am to 11:59pm)
- Tensor created from the log:
  - 5 million non-zeros
- Decomposition took ~500 seconds

Time x Source IP x Destination IP x DNS Query String x DNS Query Type

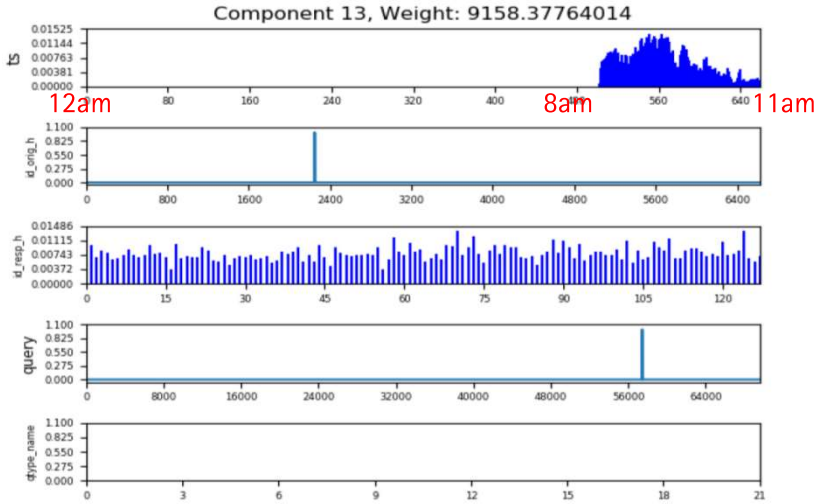
Evolution of the Attack Over Time

… as seen with streaming decompositions

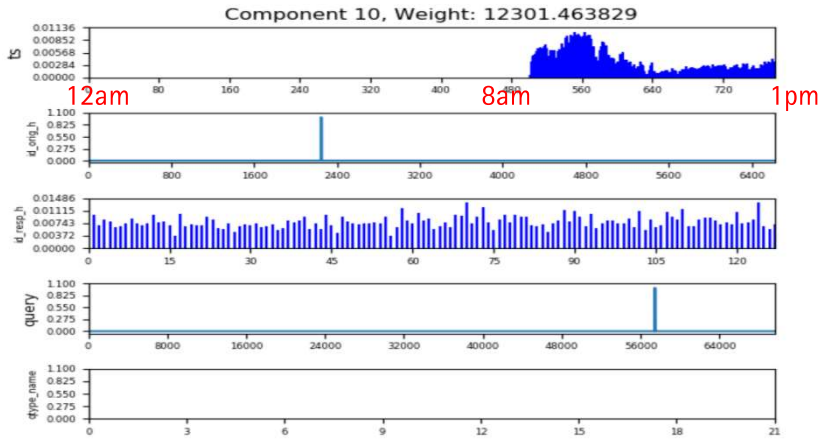
State of the activity at 9am



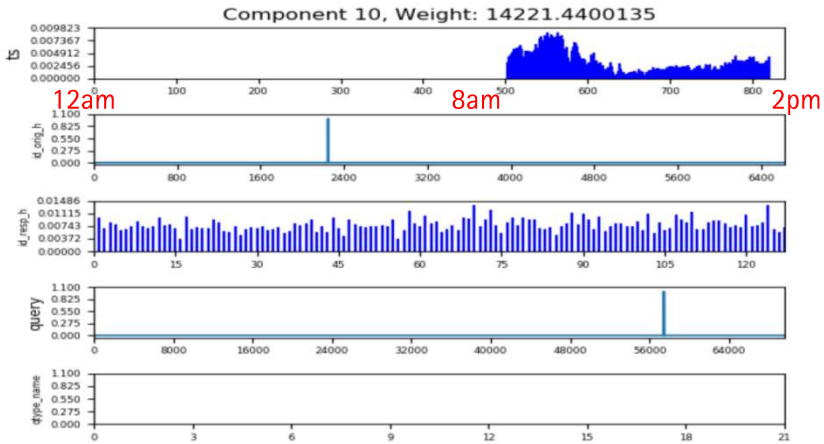
State of the activity at 11am



State of the activity at 1pm

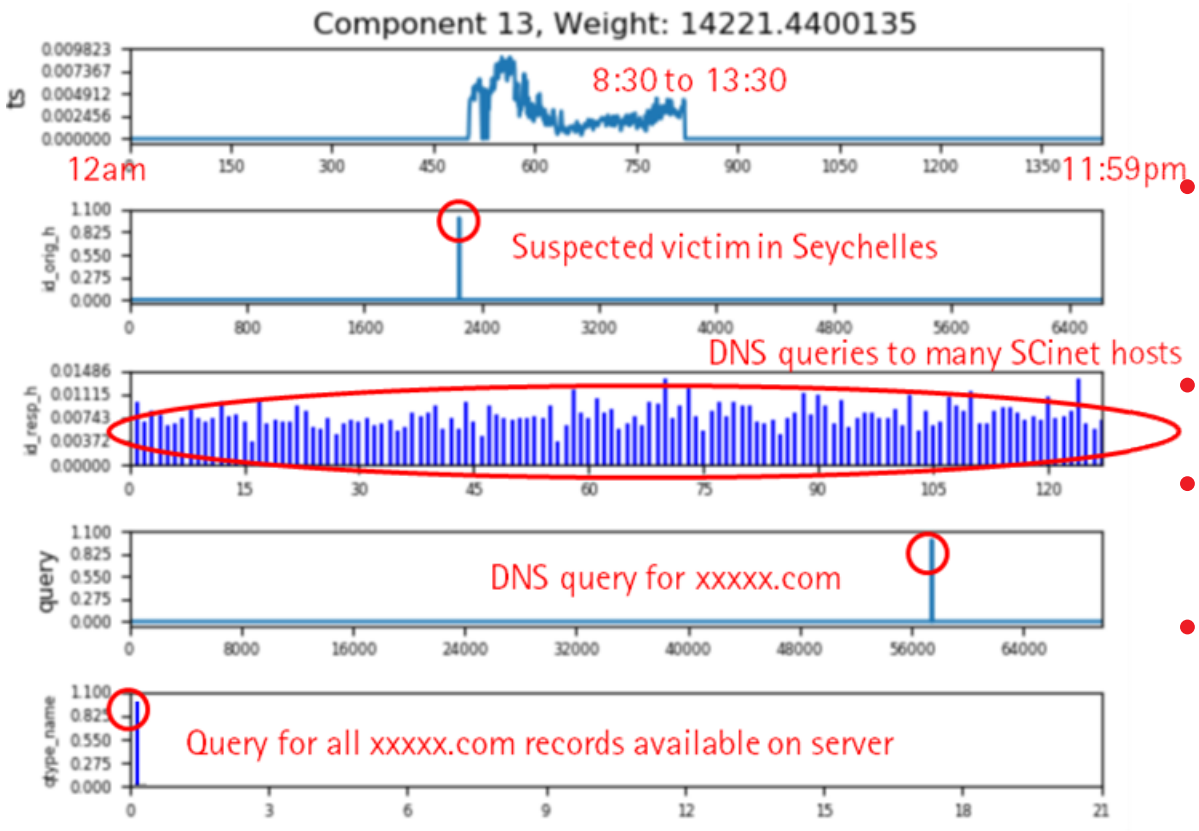


State of the activity at 2pm



DNS Amplification DDoS attack

… as seen with streaming decompositions



DNS amplification DDoS attack identified from streaming decompositions:

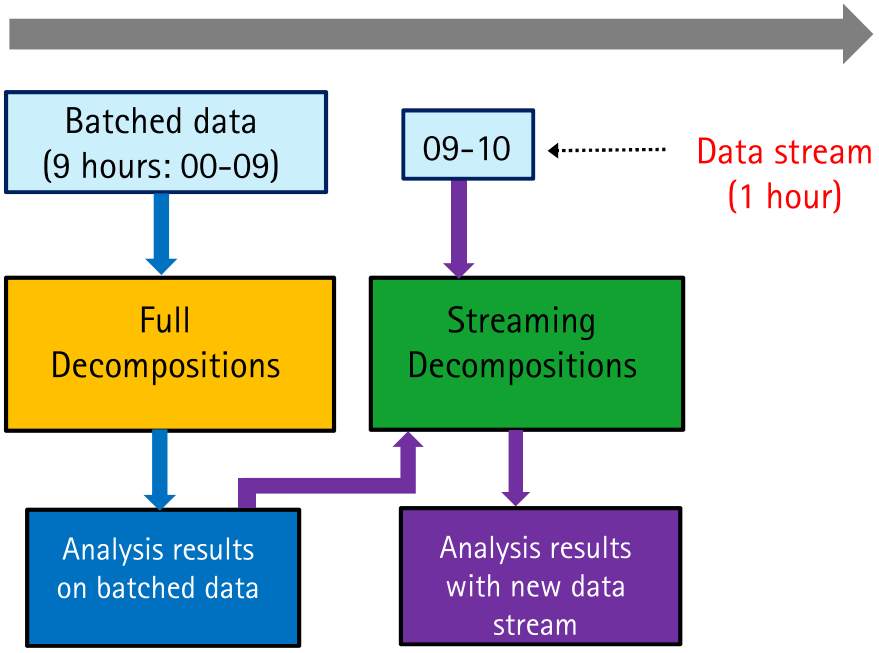
- Analyzed DNS log collected over 24 hours (12am to 11:59pm)
  - Base: 8 hours (12am to 7:59am)
  - Streams: every 1 hour (from 8am)
- Base decomposition on first 8 hours of data took ~200 seconds
- Subsequent streaming decompositions on 1-hour streams took ~3-4 seconds each
- Identified the attack at its onset in near real-time

Time x Source IP x Destination IP x DNS Query String x DNS Query Type

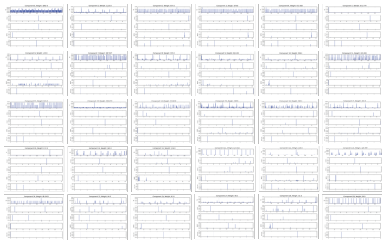
ENSIGN Streaming Decompositions

… providing useful meta-information to users

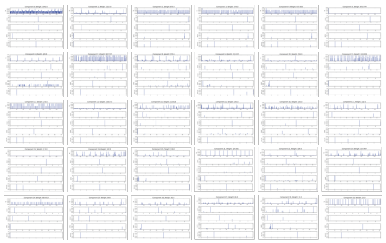
Data continuously generated over time by network sensors



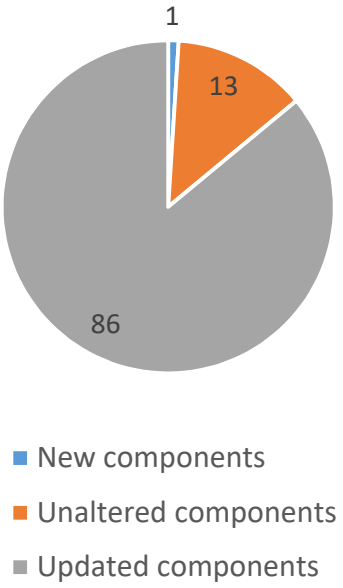
100 Base Components



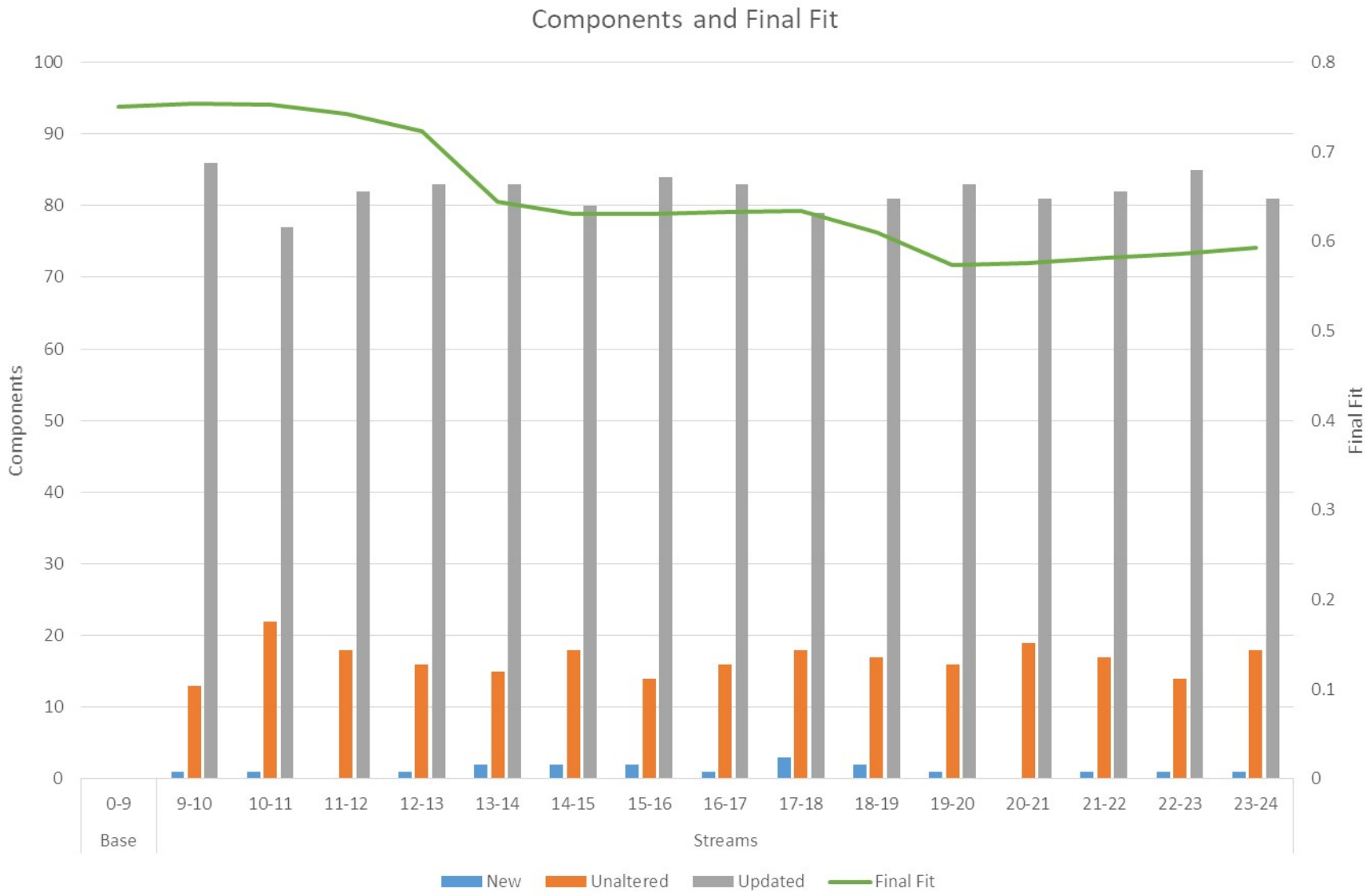
100 Components After Streaming Analysis



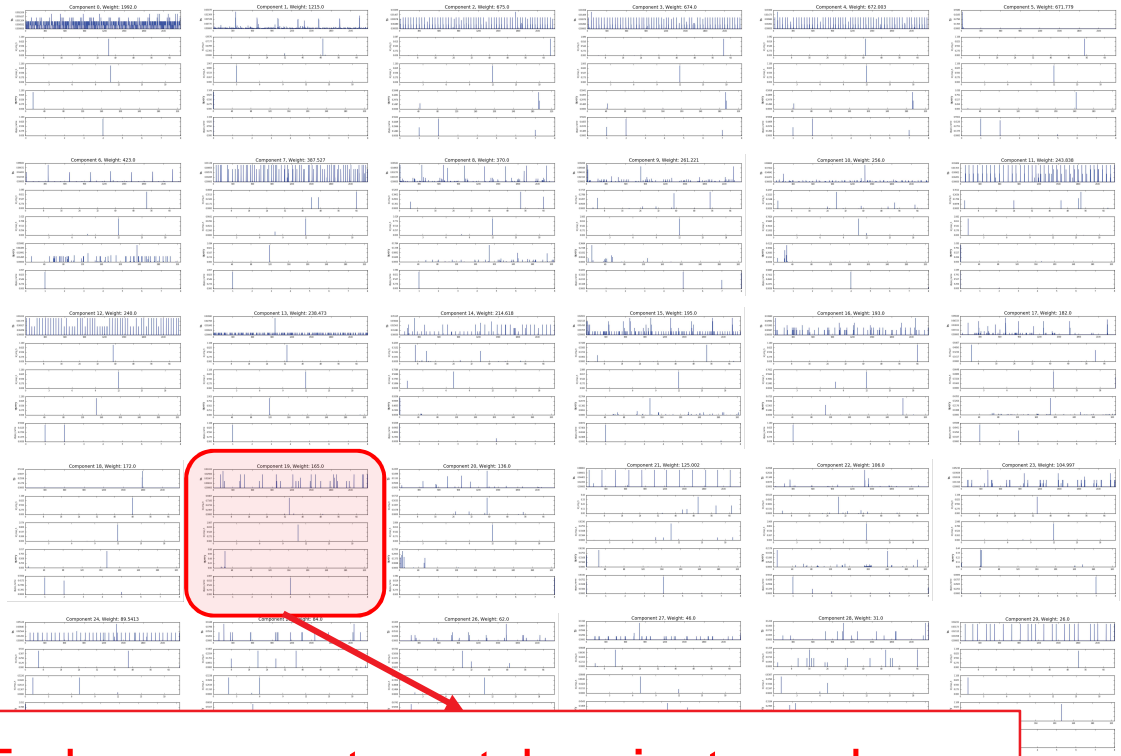
Visualizable report on how components evolve



How the Decomposition Evolves



The Barrier – Automate Component Classification



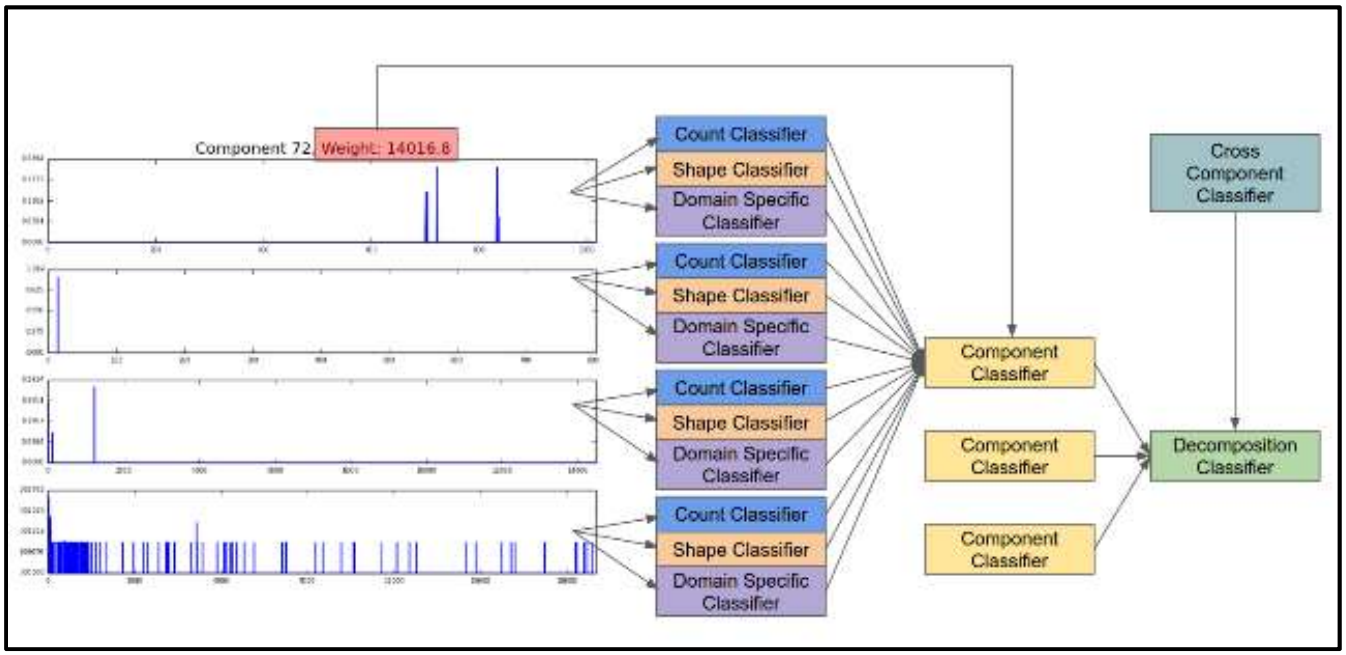
Hundreds of components needed to characterize network traffic

Most components are benign

Each component can take minutes or hours to manually investigate

Which components are interesting?

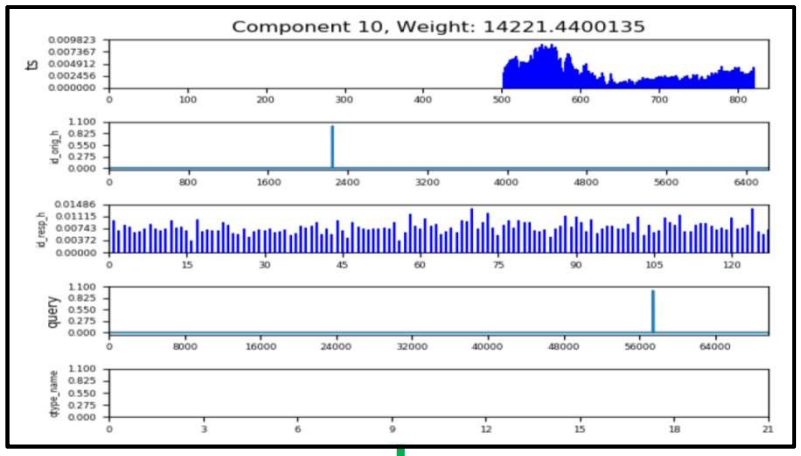
General Classification for Decompositions



Hierarchical "white box" classification of Modes, Components, and Decomposition

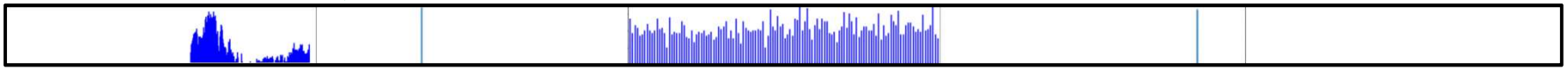
- Count Classifier
  - Finds 1-1, 1-many, many-1 relationships – port scan, network map, DNS amplification
- Shape Classifier
  - Finds recognizable patterns – periodic beaconing, after hours traffic
- Domain Specific Classifier
  - Incorporates domain knowledge – Port 22 is SSH traffic, Ports 80 and 443 are web traffic

Topic Modeling for Component Classification



Latent Dirichlet Allocation (LDA)

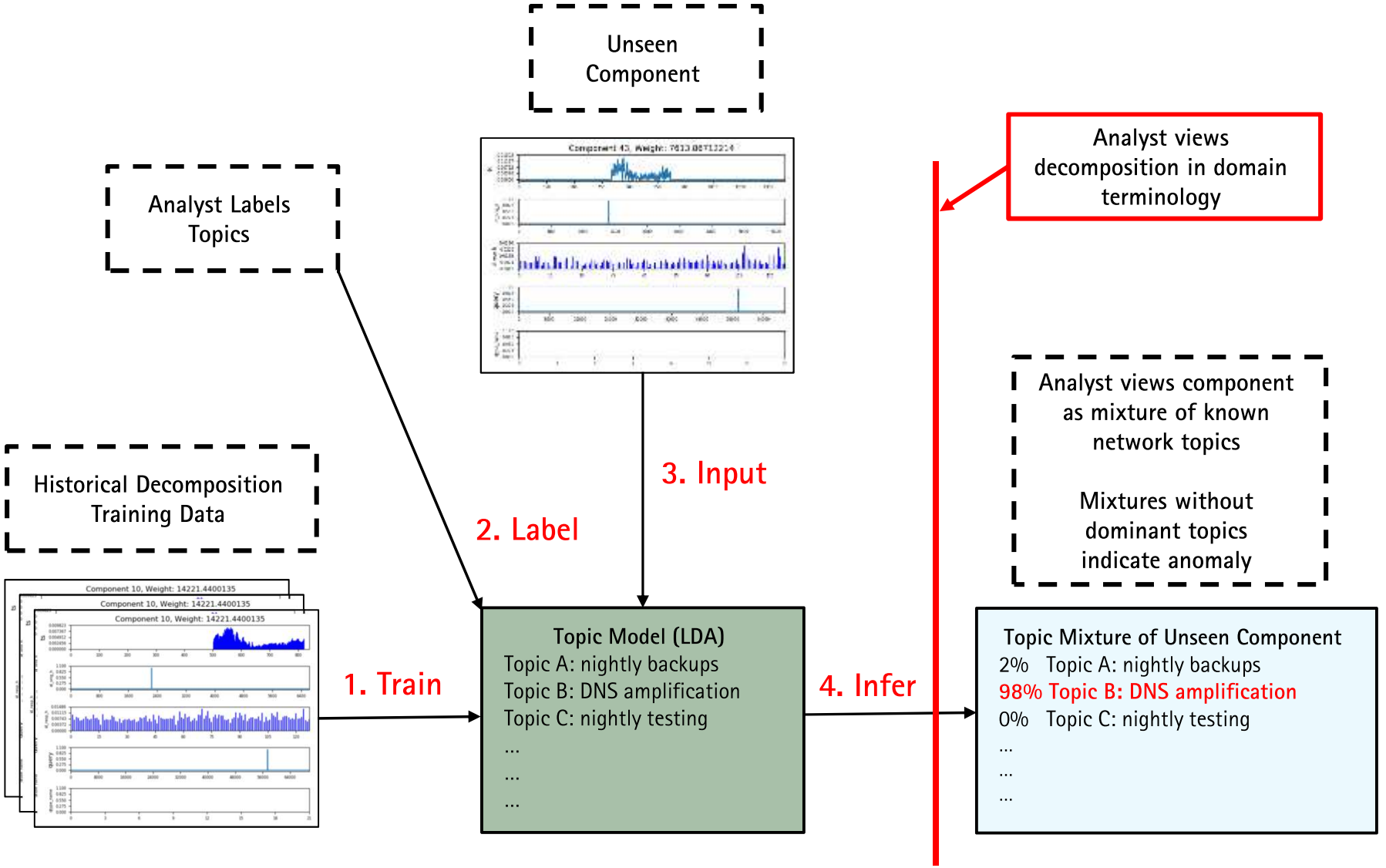
- Well-known Bayesian topic modeling algorithm
- Learns topic model from a corpus of documents
- Infers topic mixture of new documents
- Online updates of topic model
- Commonly used in other applications
  - Bioinformatics
  - Image, video, and sound processing
  - Collaborative filtering



Mapping tensor decompositions to LDA concepts

- Component (as vector) = "document"
- Label = "word"
- Score = "word count"
- Topic = recognizable pattern of network behavior

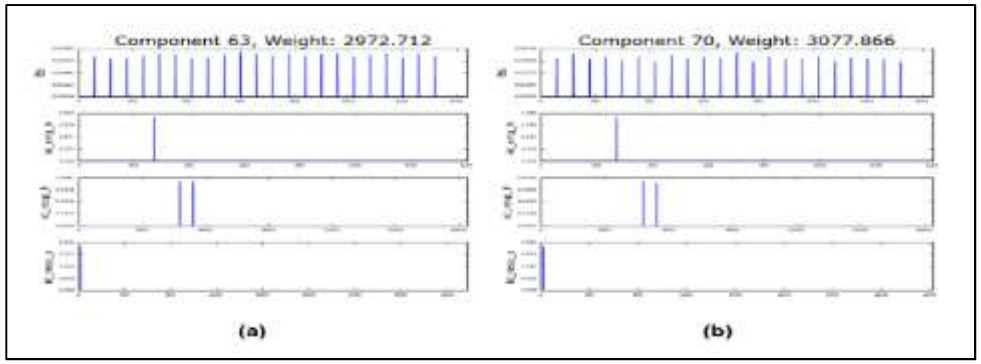
Topic Model Classification Workflow



Topic Model Classification Experiments

Small office network traffic decompositions

- Train on 5 days, test on 2 days – 100 topics
- Time, Source IP, Destination IP, Port

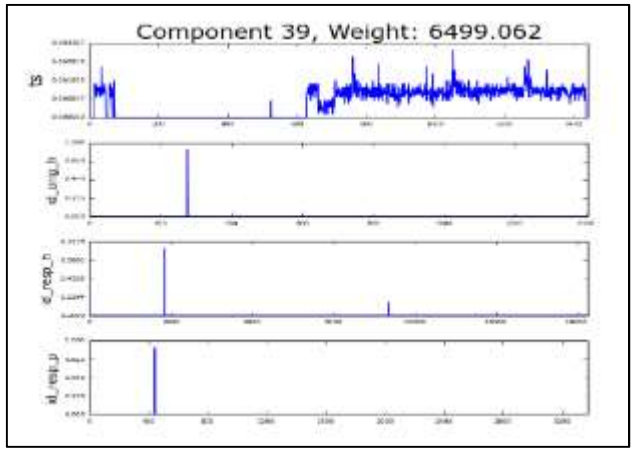


Identified well-known traffic

- Backups, log forwarding, system monitoring, …
- 100% single topic "mixtures"

Identified anomalous traffic

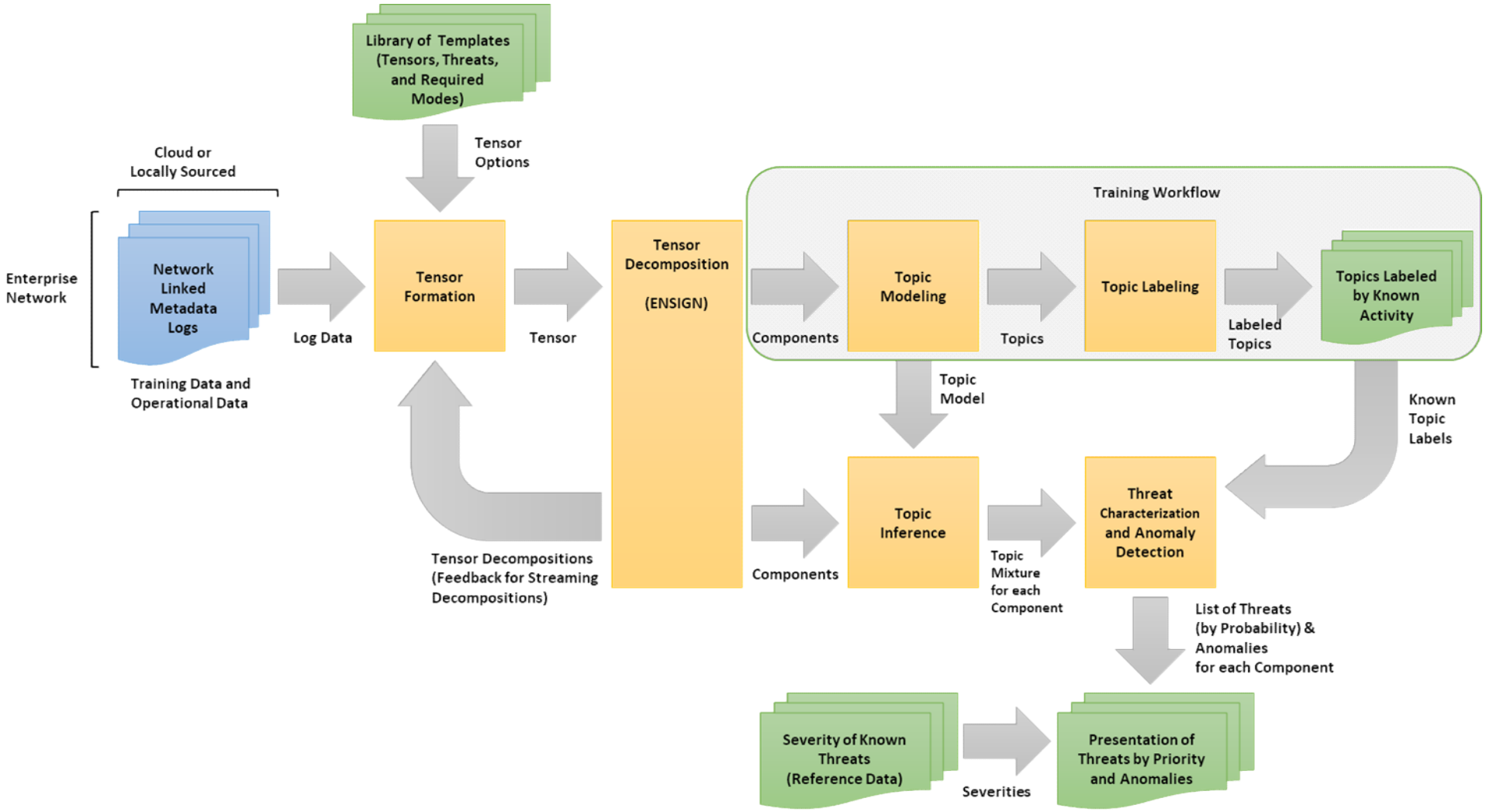
- Unusual long running VPN connection
  - 16 topic mixture
  - Largest part of topic mixture was 16%



Identified clear mixtures of topics

- 2-3 topics matched

The Barrier – Demonstrate a Repeatable Process



Tensor Creation Library

Tensor Name	Bro Log	Tensor Dimensions
Connections	The connections log (conn.log)	Time x Sender IP x Receiver IP x Port
Outgoing	Connections log entries with local sender and external receiver	Time x Sender IP x Receiver IP x Port
Incoming	Connections log entries with local receiver and external sender	Time x Sender IP x Receiver IP x Port
Time Independent	The connections log	Sender IP x Receiver IP x Port x Connection State
File Transfer	The file transfer log (files.log)	Time x Sender IP x Receiver IP x MIME-Type
HTTP	The HTTP traffic log (http.log)	Time x Sender IP x Receiver IP x URI x User Agent
DNS Query	All queries from the DNS log (dns.log)	Time x Sender IP x Receiver IP x Query x Query Type

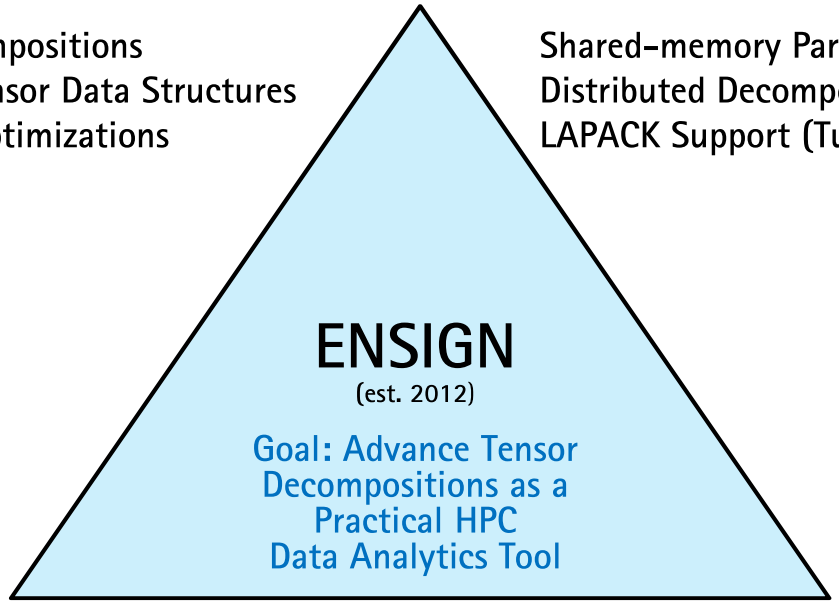
Reservoir Labs ENSIGN Project

Version:
ENSIGN 4.1 – Dec 2017

Performance

ANSI C for All Decompositions
Optimized Sparse Tensor Data Structures
Memory-efficient Optimizations

Shared-memory Parallelism (OpenMP)
Distributed Decompositions (CP-ALS, CP-APR)
LAPACK Support (Tucker)



Capability

CP-ALS
CP-ALS-NN
CP-APR
CP-APR-PDNR
CP-APR-PQNR

CP-OPT (Joint)
Tucker-ALS
CP-APR-Streaming
Tucker-ALS-LowRankUpdate

Usability

GUI & CLI
Tensor Building Tools
Visual Exploration Tools
Python Bindings
Modular Installation
Documentation
Tested for CentOS 7.2
Virtual Machine Distributions

Application Support:
Cyber Security
Geospatial Analysis
Genomics

Python Bindings & Jupyter Notebook

jupyter ENSIGN-Jupyter Last Checkpoint: 4 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Python [conda root]

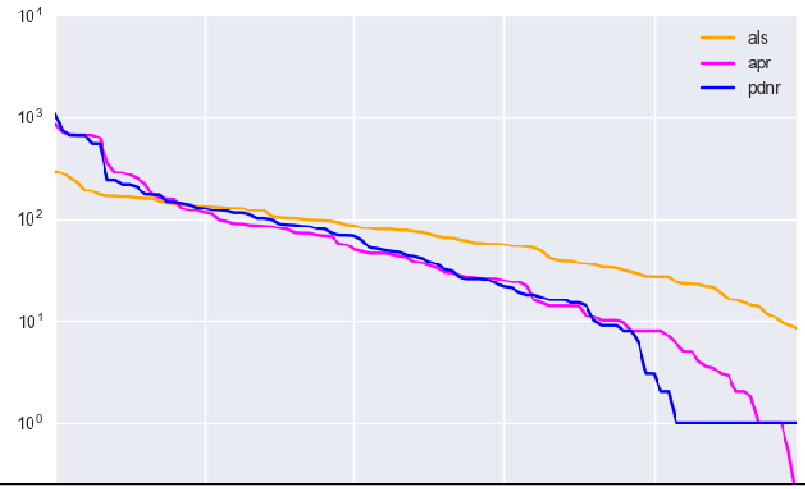
Code CellToolbar

In [2]:
```
import ensign.cp_dccomp as cpd
import ensign.sptensor as spt

# Parameters
rank = '100'
sptensor_file = 'tensor_data.txt'

# Load tensor & decompose
the_tensor = spt.read_sptensor_file(sptensor_file)
als_decomp = cpd.cp_als(the_tensor, rank, "als_save_dir")
apr_decomp = cpd.cp_apr(the_tensor, rank, "apr_save_dir")
pdnr_decomp = cpd.cp_apr_pdnr(the_tensor, rank, "pdnr_save_dir")
```

In [3]:
```
als_weights = pd.Series(als_decomp.weights)
apr_weights = pd.Series(apr_decomp.weights)
pdnr_weights = pd.Series(pdnr_decomp.weights)
```

In [4]:
```
ax = als_weights.plot(color='orange', logy=True, label='als', legend=True)
ax = apr_weights.plot(color='magenta', logy=True, label='apr', ax=ax, legend=True)
pdnr_weights.plot(color='blue', logy=True, label='pdnr', ax=ax, legend=True)
```

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x7f29bd28fe90>



Shoulders of Giants

Much of this work builds on foundational contributions in mathematics and algorithms developed over decades by the community of tensor researchers.

In particular, we would like to acknowledge the work of Tamara Kolda and the staff at Sandia National Laboratories.

http://www.sandia.gov/~tgkolda/

- Kolda, T., Bader, B., **Tensor Decompositions and Applications**, *SIAM Review, 51(3), pp. 455-500, 2009*.
- Chi, E., Kolda, T., **On Tensors, Sparsity, and Nonnegative Factorizations**, *SIAM Journal on Matrix Analysis and Applications 33.4 (2012):1272-1299*.
- Tensor Toolbox for MATLAB and C++

Conclusion

Reservoir Labs

- https://www.reservoir.com

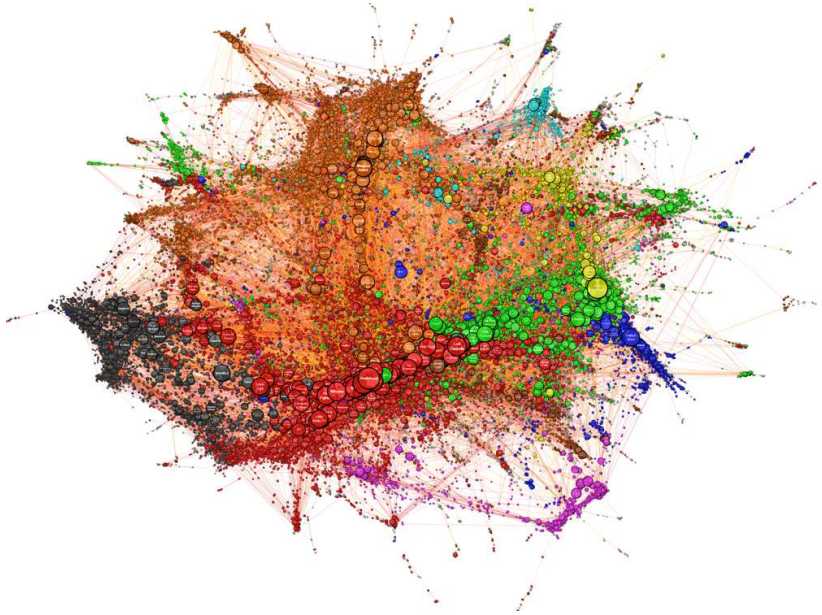
Contact the Speaker

- ezick@reservoir.com

Recent Papers

- **Enhancing Network Visibility through Tensor Analysis**
  M. Baskaran, T. Henretty, D. Bruns-Smith, J. Ezick, R. Lethin in SC17 INDIS Workshop, Nov 2017

- **Memory-efficient Parallel Tensor Decompositions**
  M. Baskaran, T. Henretty, B. Pradelle, M. H. Langston, D. Bruns-Smith, J. Ezick, R. Lethin in IEEE HPEC, Sep 2017 (Best Paper Award)



Tensor decompositions provide a fast, scalable, practical linear algebra based solution to finding patterns in linked metadata