Uncovering Priority Anomalies Using Pattern Discovery as a Roadmap for Contextual Analysis

Thomas Henretty
henretty@reservoir.com

FloCon 2020
Savannah, GA
9 January 2018

Reservoir Labs
New York, NY
www.reservoir.com

Presentation Outline

Part 1: Background

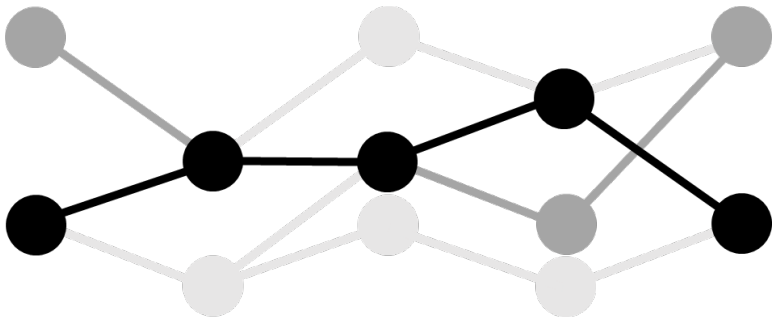
- Tensor Decomposition Basics
- Pattern Discovery in Network Flows
- MITRE ATT&CK Framework

Part 2: Anomaly Ranking

- Decompositions as Documents
- Topic Modeling for Anomaly Ranking
- Other Techniques

Part 3: Graphs and Databases

- Constructing a Targeted Query



Pattern Discovery

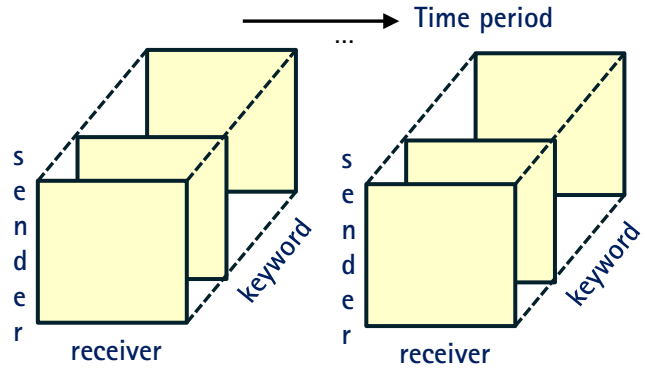
Tensor decomposition provides a model for Zeek log data that allows behaviors to be separated as coherent patterns

PART 1: BACKGROUND

Tensors: Representing Multidimensional Data

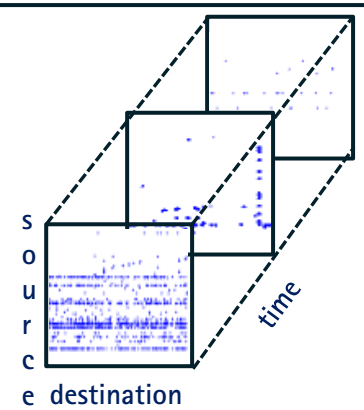
Real World Data

- Multidimensional
- Heterogeneous
- Large
- Sparse



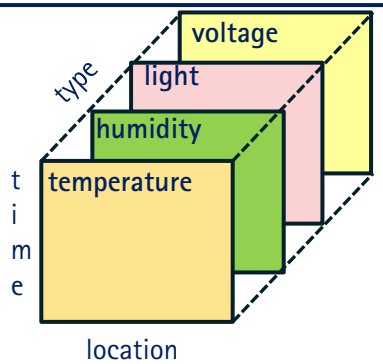
Sender x Receiver x Keyword x Time period

Email Data



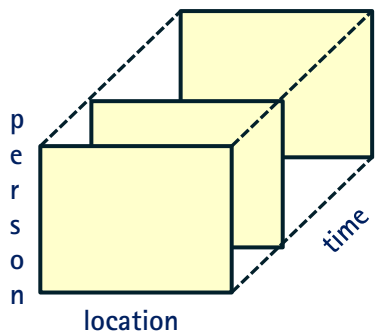
Src x Dest x Time

Network Traffic Data



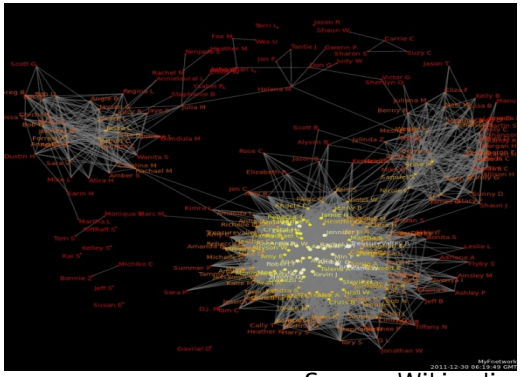
Time x Location x Type

Environmental Sensor Monitoring



Person x Location x Time

Physical Access Data

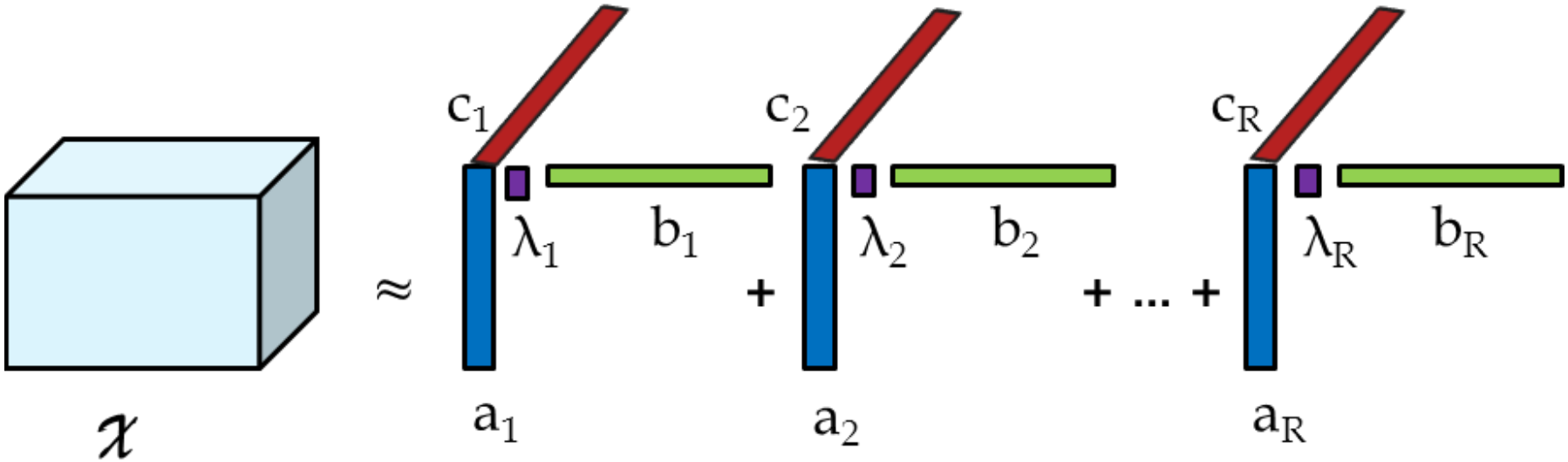


Source: Wikipedia

Person x Person x Relation

Social Network Graph

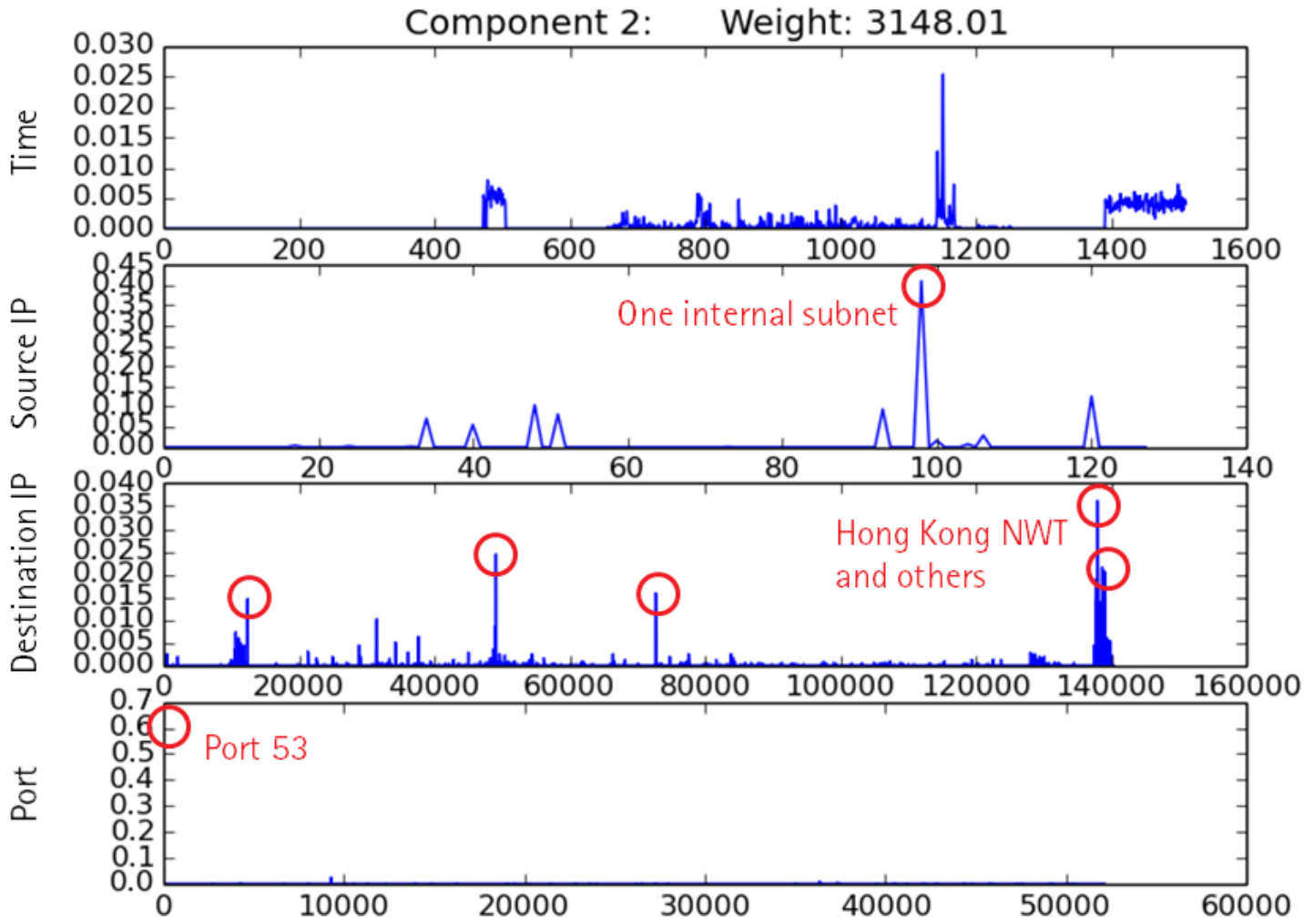
Basic CP Tensor Decomposition



CP tensor decomposition

- Multidimensional analog to matrix factorization
- Break tensor into *R* components
- Components represent correlated data (quantitatively)
- Can reconstruct tensor from subset of components

Example Component: Suspicious DNS Traffic



Time x Source IP x Destination IP x Port

Tensor Library for Cybersecurity

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

Tensor Decompositions in MITRE ATT&CK

Relevant techniques in the MITRE ATT&CK framework

- Depends on data decomposed
- Focus on network flows
  - Netflow – Techniques detected via Netflow/Enclave Netflow
  - Zeek logs – Netflow + Network Protocol Analysis + Network Intrusion Detection

Relevant tactics

- When decomposing Zeek logs …
  - Initial Access (3 of 11 techniques)
  - Execution (3 of 34)
  - Persistence (5 of 62)
  - Privilege Escalation (1 of 32)
  - Defense Evasion (5 of 69)
  - Credential Access (3 of 21)
  - Discovery (4 of 23)
  - Lateral Movement (4 of 18)
  - Collection (0 of 13)
  - Command and Control (20 of 22)
  - Exfiltration (3 of 9)
  - Impact (4 of 16)

Substantially increase coverage by adding host data (e.g., Sysflow, Event Log, …)

Tensor Decomposition Coverage in ATT&CK

Covered: Data can be converted to tensors, decomposed, and anomalies identified

Initial Access 11 items	Execution 34 items	Persistence 62 items	Privilege Escalation 32 items	Defense Evasion 69 items	Credential Access 21 items	Discovery 23 items	Lateral Movement 18 items	Collection 13 items	Command And Control 22 items	Exfiltration 9 items	Impact 16 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Destruction
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Component Object Model and Distributed COM	Clipboard Data	Connection Proxy	Data Encrypted	Data Encrypted for Impact
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Clear Command History	Credentials from Web Browsers	File and Directory Discovery	Internal Spearphishing	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Disk Structure Wipe
Spearphishing Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	Code Signing	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Endpoint Denial of Service
Spearphishing via Service	Execution through API	BITS Jobs	Dylib Hijacking	Compile After Delivery	Exploitation for Credential Access	Network Sniffing	Pass the Ticket	Data Staged	Domain Fronting	Exfiltration Over Physical Medium	Firmware Corruption
Supply Chain Compromise	Execution through Module Load	Bootkit	Elevated Execution with Prompt	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Inhibit System Recovery
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Emond	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Input Capture	Fallback Channels		Network Denial of Service
Valid Accounts	Graphical User Interface	Change Default File Association	Exploitation for Privilege Escalation	Component Object Model Hijacking	Input Capture	Permission Groups Discovery	Remote Services	Man in the Browser	Multi-hop Proxy		Resource Hijacking
	InstallUtil	Component Firmware	Extra Window Memory Injection	Connection Proxy	Input Prompt	Process Discovery	Replication Through Removable Media	Screen Capture	Multi-Stage Channels		Runtime Data Manipulation
	Launchctl	Component Object Model Hijacking	File System Permissions Weakness	Control Panel Items	Kerberoasting	Query Registry	Shared Webroot	Video Capture	Multiband Communication		Service Stop
	Local Job Scheduling	Create Account	Hooking	DCShadow	Keychain	Remote System Discovery	SSH Hijacking		Multilayer Encryption		Stored Data Manipulation
	LSASS Driver	DLL Search Order Hijacking	Image File Execution Options Injection	Deobfuscate/Decode Files or Information	LLMNR/NBT-NS Poisoning and Relay	Security Software Discovery	Taint Shared Content		Port Knocking		System Shutdown/Reboot
	Mshta	Dylib Hijacking	Launch Daemon	Disabling Security Tools	Network Sniffing	Software Discovery	Third-party Software		Remote Access Tools		Transmitted Data Manipulation
	PowerShell	Emond	New Service	DLL Search Order Hijacking	Password Filter DLL	System Information Discovery	Windows Admin Shares		Remote File Copy		
	Regsvcs/Regasm	External Remote Services	Parent PID Spoofing	DLL Side-Loading	Private Keys	System Network Configuration Discovery	Windows Remote Management		Standard Application Layer Protocol		
	Regsvr32	File System Permissions Weakness	Path Interception	Execution Guardrails	Securityd Memory	System Network Connections Discovery			Standard Cryptographic Protocol		
	Rundll32	Hidden Files and Directories	Plist Modification	Exploitation for Defense Evasion	Steal Web Session Cookie	System Owner/User Discovery			Standard Non-Application Layer Protocol		
	Scheduled Task	Hooking	Port Monitors	Extra Window Memory Injection	Two-Factor Authentication Interception	System Service Discovery			Uncommonly Used Port		
	Scripting	Hypervisor	PowerShell Profile	File and Directory Permissions Modification		System Time Discovery			Web Service		
	Service Execution	Image File Execution Options Injection	Process Injection	File Deletion		Virtualization/Sandbox Evasion					
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Scheduled Task	File System Logical Offsets							
	Signed Script Proxy Execution	Launch Agent	Service Registry Permissions Weakness	Gatekeeper Bypass							
	Source	Launch Daemon	Setuid and Setgid	Group Policy Modification							
	Space after Filename	Launchctl	SID-History Injection	Hidden Files and Directories							
	Third-party Software	LC_LOAD_DYLIB Addition	Startup Items	Hidden Users							
	Trap	Local Job Scheduling	Sudo	Hidden Window							
	Trusted Developer Utilities	Login Item	Sudo Caching	HISTCONTROL							
	User Execution	Logon Scripts	Valid Accounts	Image File Execution Options Injection							
	Windows Management Instrumentation	LSASS Driver	Web Shell	Indicator Blocking							
	Windows Remote Management	Modify Existing Service		Indicator Removal from Tools							
	XSL Script Processing	Netsh Helper DLL		Indicator Removal on Host							
		New Service		Indirect Command Execution							
		Office Application Startup		Install Root Certificate							
		Path Interception		InstallUtil							
		Plist Modification		Launchctl							
		Port Knocking		LC_MAIN Hijacking							
		Port Monitors		Masquerading							
		PowerShell Profile		Modify Registry							
		Rc.common		Mshta							
		Re-opened Applications		Network Share Connection Removal							
		Redundant Access		NTFS File Attributes							
		Registry Run Keys / Startup Folder		Obfuscated Files or Information							
		Scheduled Task		Parent PID Spoofing							
		Screensaver		Plist Modification							
		Security Support Provider		Port Knocking							
		Server Software Component		Process Doppelgänging							
		Service Registry Permissions Weakness		Process Hollowing							
		Setuid and Setgid		Process Injection							
		Shortcut Modification		Redundant Access							
		SIP and Trust Provider Hijacking		Regsvcs/Regasm							
		Startup Items		Regsvr32							
		System Firmware		Rootkit							
		Systemd Service		Rundll32							
		Time Providers		Scripting							
		Trap		Signed Binary Proxy Execution							
		Valid Accounts		Signed Script Proxy Execution							
		Web Shell		SIP and Trust Provider Hijacking							
		Windows Management Instrumentation Event Subscription		Software Packing							
		Winlogon Helper DLL		Space after Filename							
				Template Injection							
				Timestomp							
				Trusted Developer Utilities							
				Valid Accounts							
				Virtualization/Sandbox Evasion							
				Web Service							
				XSL Script Processing							

Covered by Zeek log tensor decompositions

Covered by host data tensor decompositions

Example Detection of ATT&CK Technique

Tactic and Technique

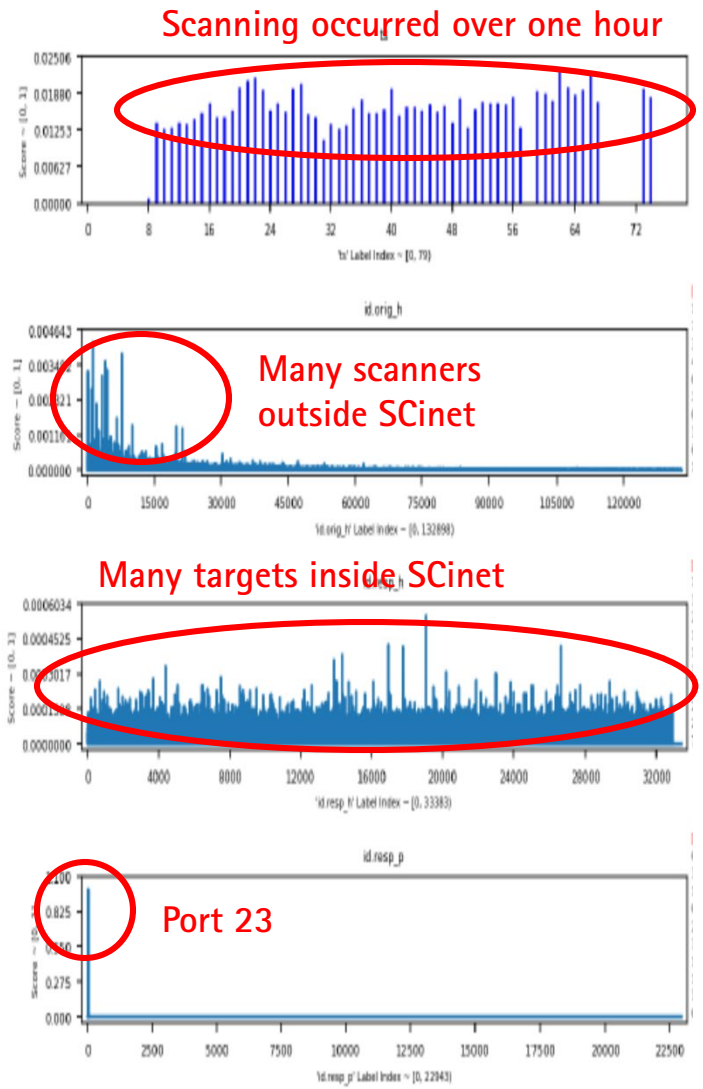
- Discovery – Network Service Scanning

Context

- SCinet 2019
- Network for Supercomputing conference
- All IP addresses public (no firewalls)
- No authentication / authorization
- ~8 Million flows per hour

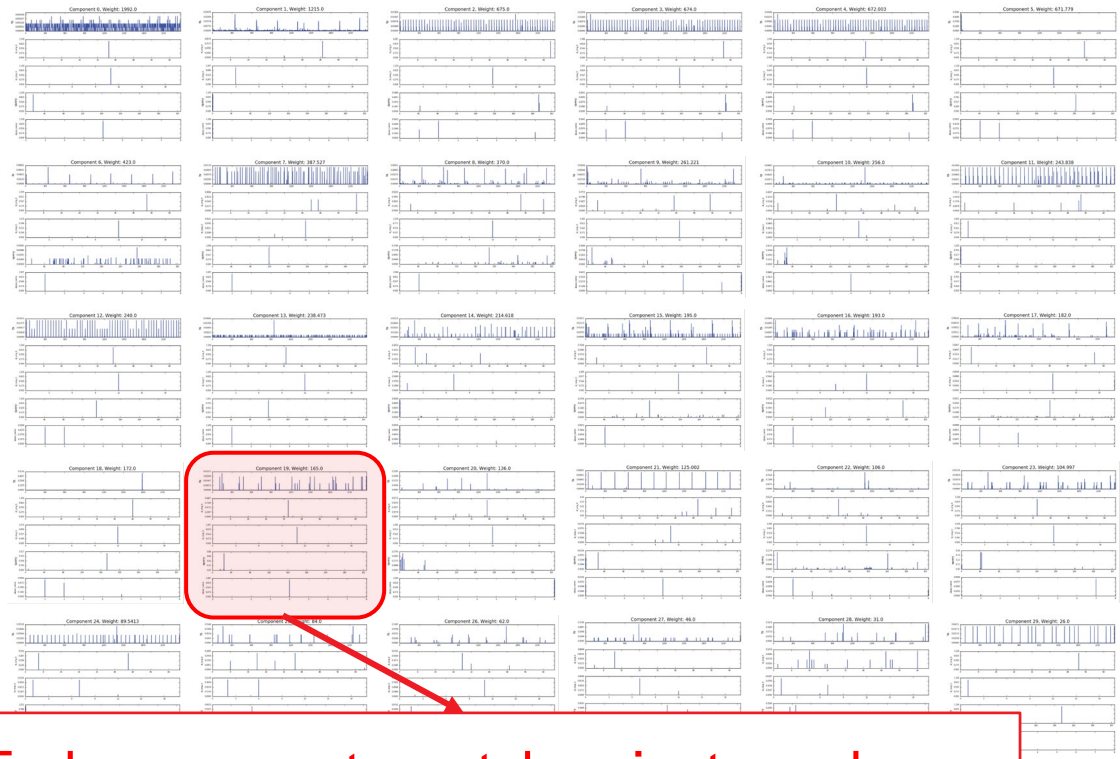
Details

- Large number of external hosts scanning SCinet
- ~176K flows on port 23
- Potential coordination
- Scan evaded other scan detection tools



PART 2: ANOMALY DETECTION

Need to Automate Anomaly Detection



Each component can take minutes or hours to manually investigate

Which components are interesting?

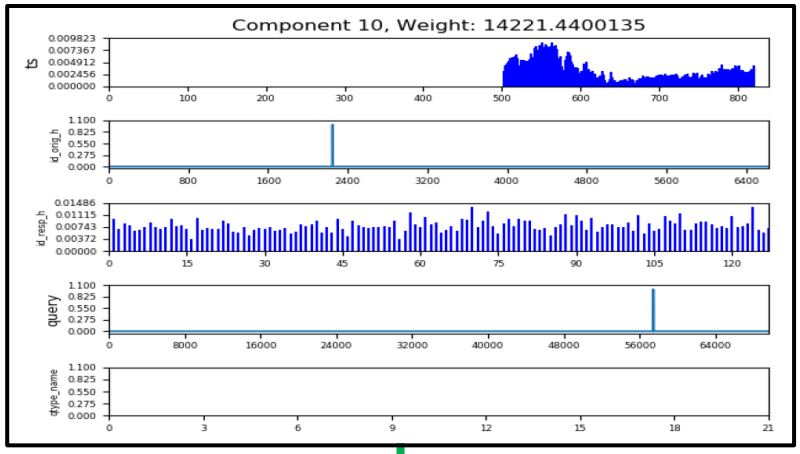
Often 100+ components needed to characterize network traffic

Most components are benign

Challenge is to identify and rank components representing anomalous behavior

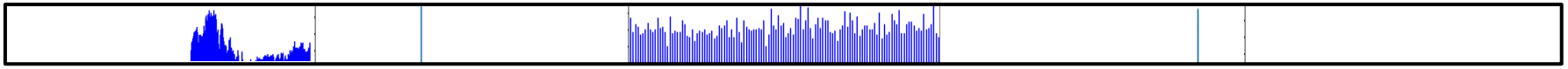
Components are trailheads for further investigation

Topic Modeling for Component Classification



Latent Dirichlet Allocation (LDA)

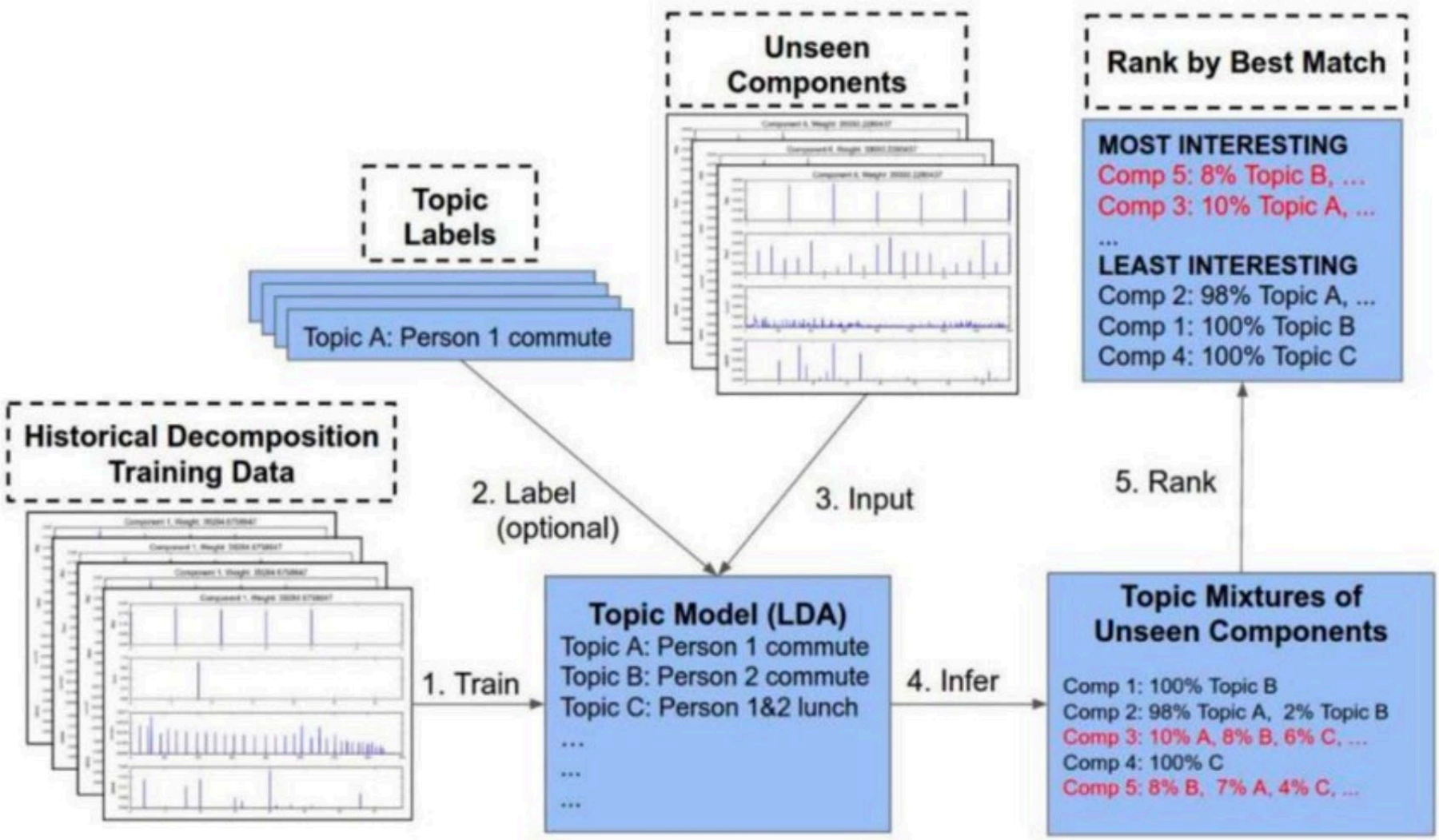
- Well-known Bayesian topic modeling algorithm
- Learns topic model from a corpus of documents
- Infers topic mixture of new documents
- Online updates of topic model
- Commonly used in other applications
  - Bioinformatics
  - Image, video, and sound processing
  - Collaborative filtering



Mapping tensor decompositions to LDA concepts

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

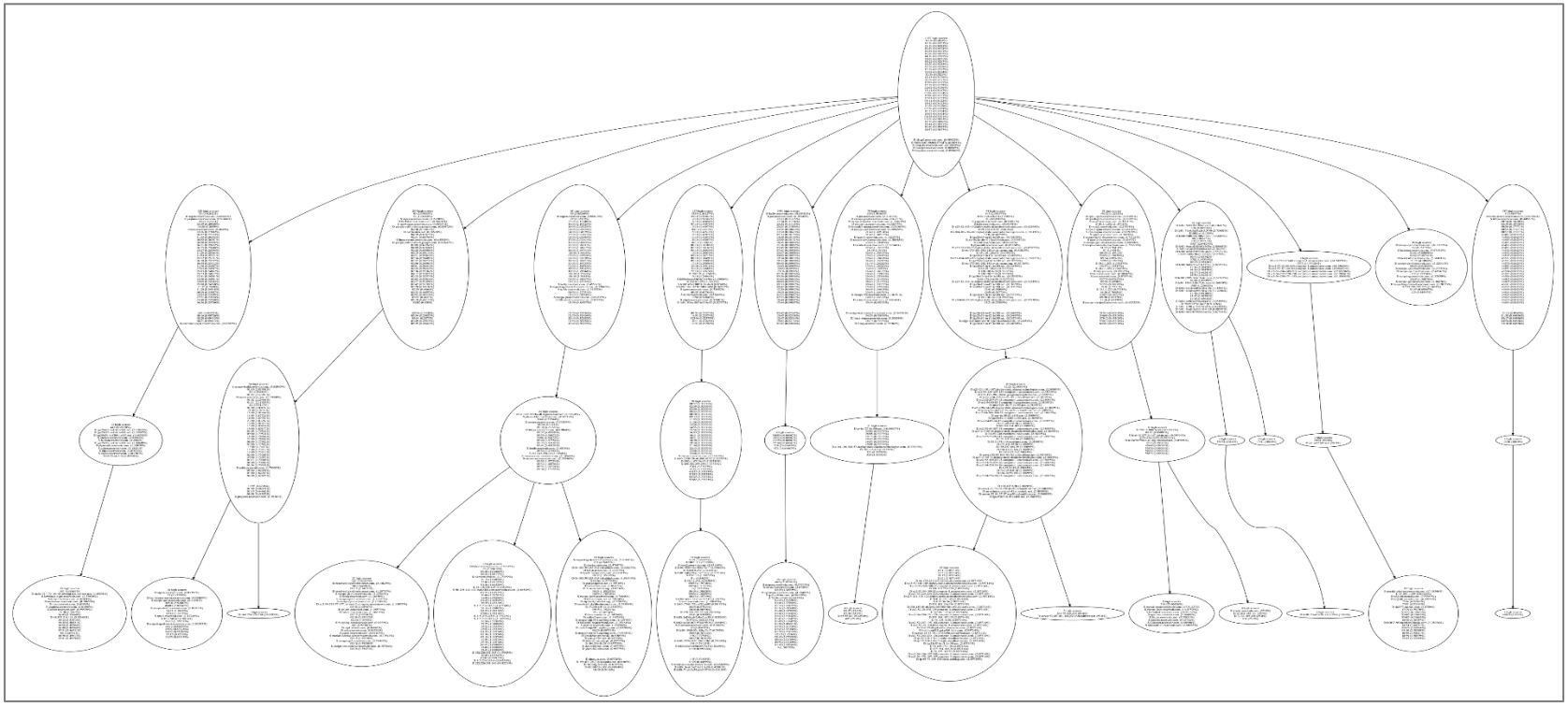
LDA Dominant Topic Approach



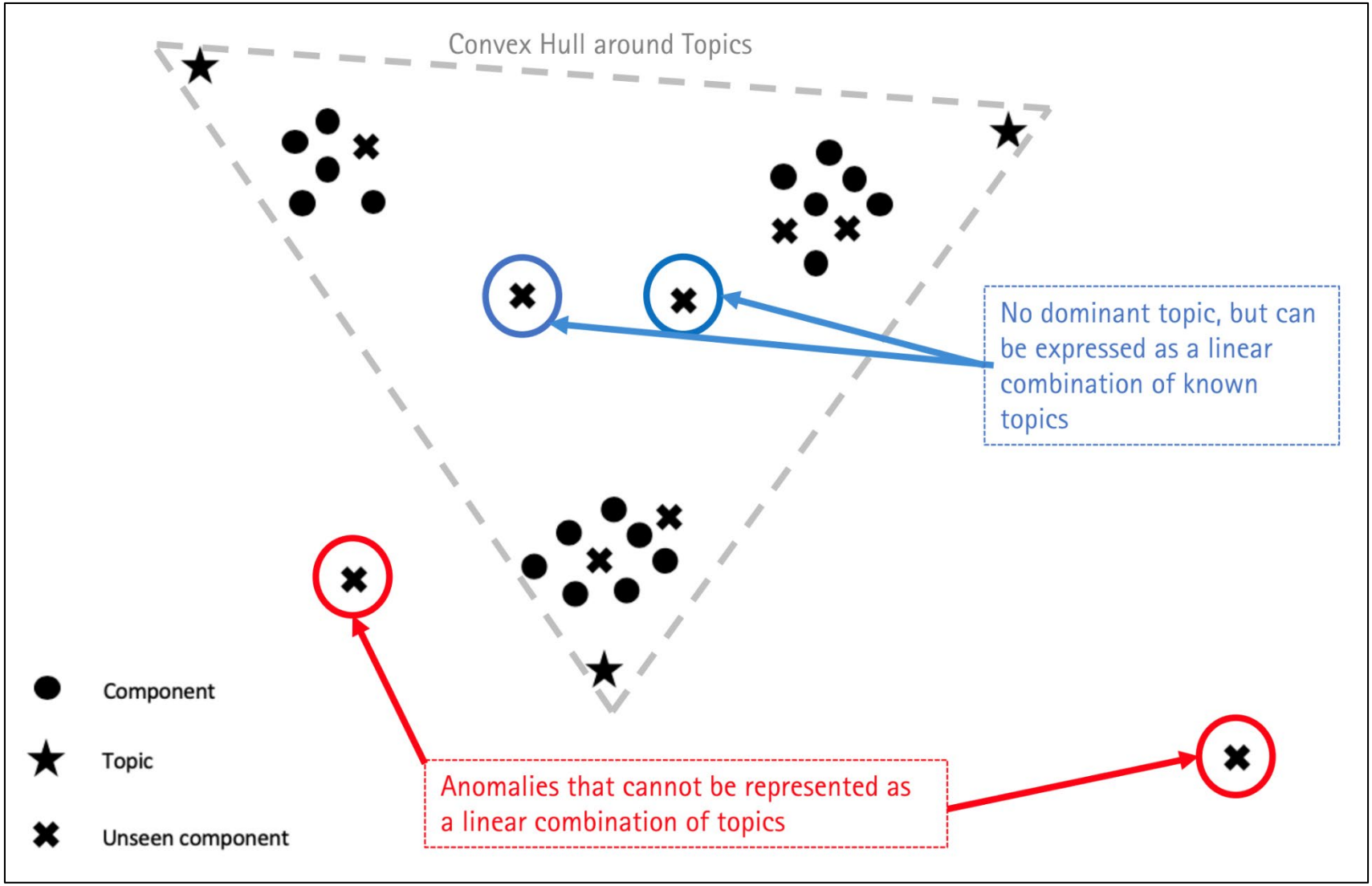
Hierarchical LDA Approach

Learn topics in tree

- Coarse grain behavior at root, fine grain at leaves
- Topic is weighted mixture of root-to-leaf paths in tree
- Same approach as dominant topic otherwise



Limitations of Dominant Topic Approaches



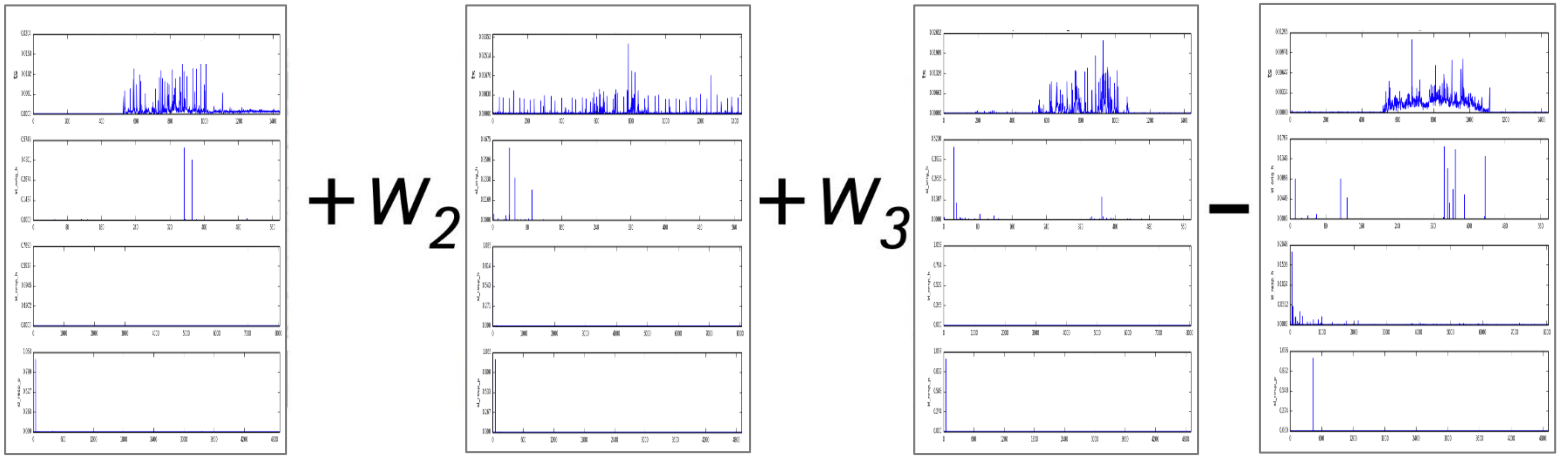
Component Reconstruction Approach

Addresses mathematical limitations of dominant topic approach

Infer topic mixtures for unseen components and reconstruct with known topics

Compare to unseen component and rank by reconstruction error

$$\left\| w_1 \; + w_2 \; + w_3 \; - \; \right\|^2$$



Topic 1 Topic 2 Topic 3 Unseen component

Decomposition Difference Approach

Compute similarity matrix between current and historical decomposition components

Component(s) dissimilar to every historical component represents anomalous behavior

Rank by max similarity

- Unseen component matches historical component
- Unseen component does not match any historical component

Unseen Components \ Historical Components				
.00	.01	.04	.01	.99
.95	.02	.01	.00	.02
.00	.01	.00	.00	.03
.02	.98	.05	.03	.01
.00	.02	.01	.97	.01

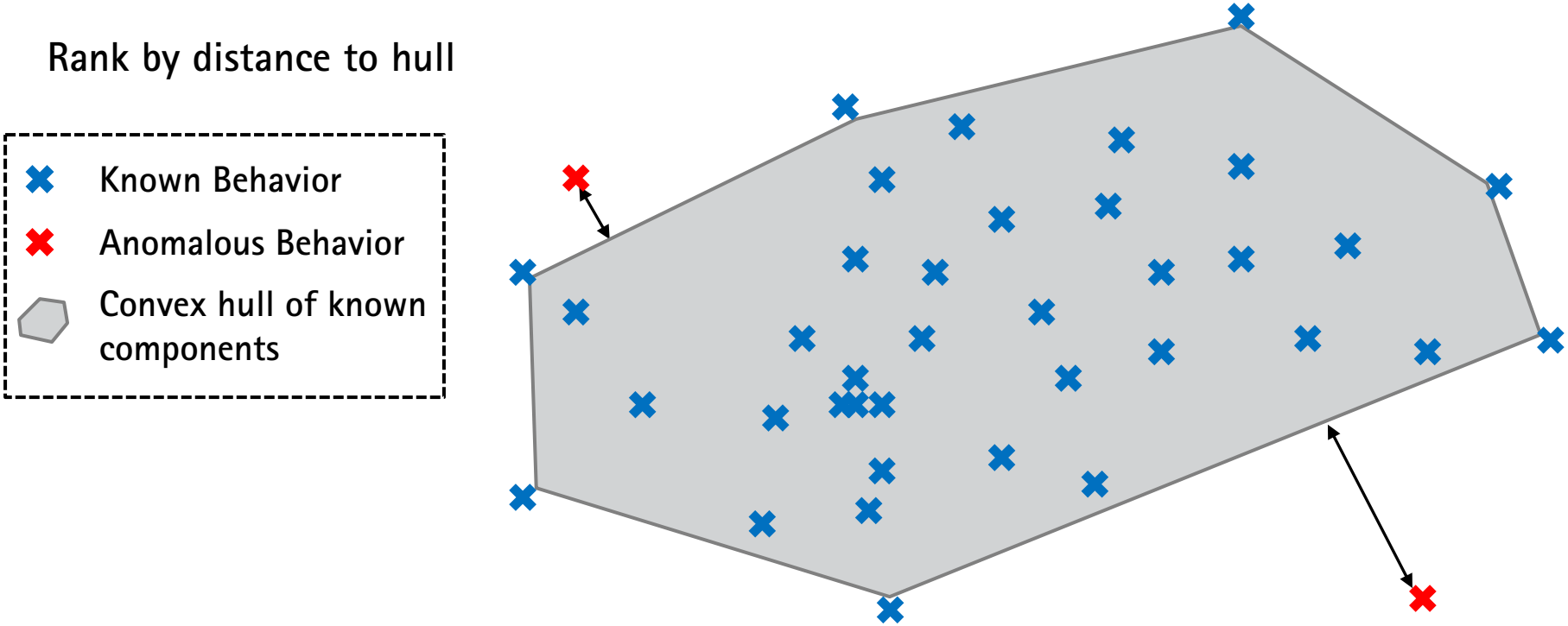
Approximate Convex Hull Approach

Compute approximate convex hull of historical decomposition components

If a component is a linear combination of historical components, it's inside the hull and we've seen all aspects of the behavior it represents

Identify anomalous components outside hull, compute distance to hull

Rank by distance to hull

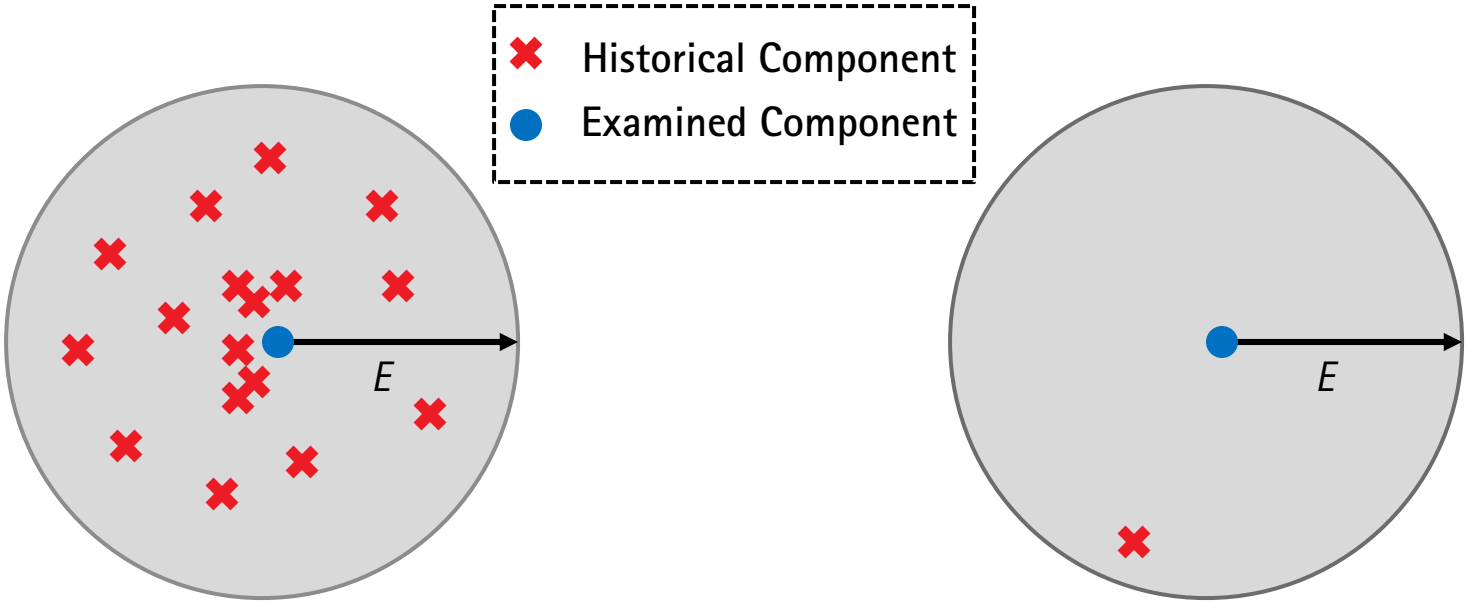


Epsilon Ball Approach

Treat component as vector, compare to historical components

Count components inside a hypersphere of radius *E*

Rank by count of components inside hypersphere



Known Behavior

Anomalous Behavior

Comparison of Anomaly Detection Approaches

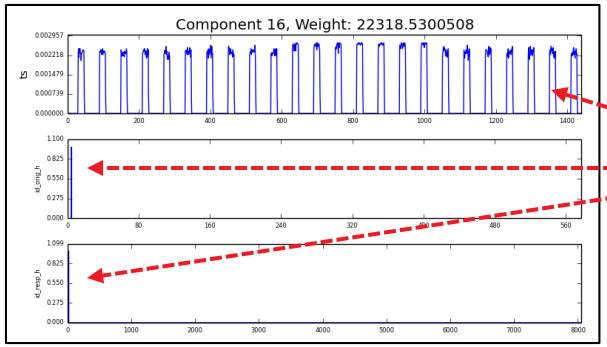
	Execution Time	Parametric	Detects Anomalous Variations of Historical Behavior	Detects Anomalous Behavior Unrelated to Historical Behavior
LDA – Dom Topic	High	Yes	Yes	No
HLDA – Dom Topic	High	No	Yes	No
LDA – Component Reconstruct	High	Yes	Yes	Yes
HLDA – Component Reconstruct	High	No	Yes	Yes
Decomp Diff	Low	Yes	Somewhat	Yes
Approximate Convex Hull	Low	No	No	Yes
Epsilon Ball	Low	Yes	Somewhat	Yes

PART 3: GRAPHS AND DATABASES

Graphs and Databases in Context

Components only tell a small part of the story

- E.g., Timestamp, Source IP, Destination IP



Component represents beaconing behavior between two IP addresses. Is it C2 traffic? Hourly batch jobs? Hourly log transfers?

More information necessary to make a malicious / benign decision

- E.g., user, asset type, network topology, known behaviors, threat intel, ...
- Needed info stored in external DB / graph / ... or enriched data in SIEM

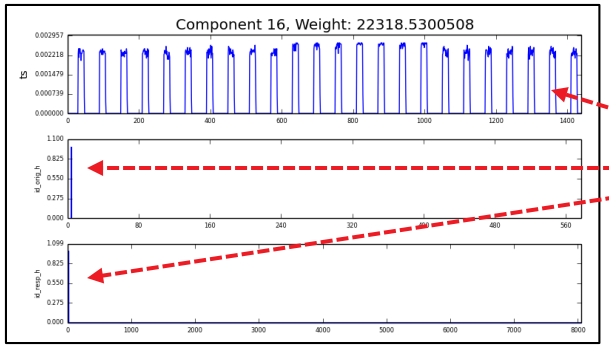
Use anomalous component as trailhead into investigation

- Generate targeted queries to provide context and assist decision making
- Massively reduces scope of graph / database analysis

Generating Targeted Queries

Use component labels with nonzero scores to generate "WHERE" clause

- E.g., "SELECT * WHERE ts=(00:00, 01:00, ...), src_ip=1.2.3.4, dst_ip=5.6.7.8"



Component represents beaconing behavior between two IP addresses. Is it C2 traffic? Hourly batch jobs? Hourly log transfers?

Problem: Data was binned before conversion to tensor

Solution Part 1: Generate backtracking data when building tensor

- Map tensor entries to lines in original log

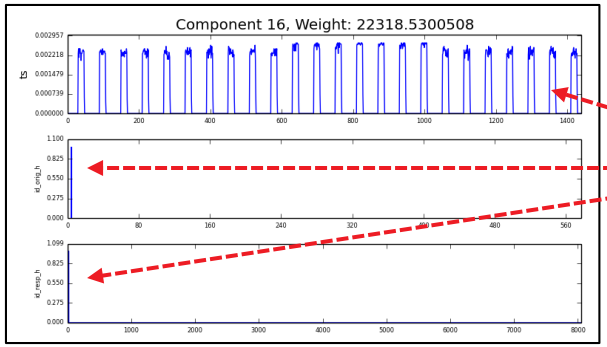
Solution Part 2: Reconstruct into tensor, get subset of relevant log entries

- Original entries provide more context – exact timestamps, flow IDs, ...

Generating Targeted Queries

Use enriched data to filter false positives

- E.g., "SELECT * WHERE ts=(00:00, 01:00, ...), src_ip=1.2.3.4, dst_ip=5.6.7.8"
  AND src_ip NOT "batch_server" AND src_ip NOT "log_transfer_hourly"



Component represents beaconing behavior between two IP addresses. Is it C2 traffic? Hourly batch jobs? Hourly log transfers?

Further queries based on results of targeted query

- Query within the returned data or use as guide for further focused queries

Targeted query massively reduces size of graph / DB / SIEM data to investigate

- Not "boiling the ocean" by running analytics over entire graph / DB / SIEM
- Tensor decompositions highly optimized and run on ten-billion scale logs in reasonable time (high minutes / low hours)

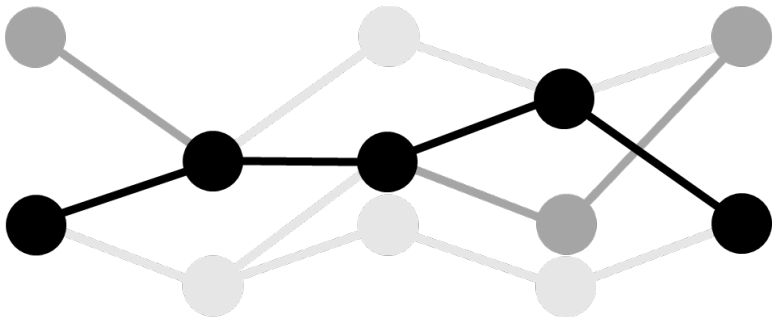
Conclusion

Contact the Speaker

- Thomas Henretty
  henretty@reservoir.com

Recent Papers

- *Combining Tensor Decompositions and Graph Analytics to Provide Cyber Situational Awareness at HPC Scale*
  HPEC, Sep 2019
- *Fast and Scalable Distributed Tensor Decompositions*
  HPEC, Sep 2019
- *Enhancing Network Visibility and Security through Tensor Analysis*
  Future Generation Computer Systems, July 2019



Pattern Discovery

Tensor decomposition provides a model for Zeek log data that allows behaviors to be separated as coherent patterns