S9385 AI-Based Anomaly Detections and Threat Forecasting for Unified Communications Networks

Kevin Riley – CTO, Ribbon

Tim Thornton - Director Software Engineering, Ribbon

About Ribbon

Ribbon is a global leader in secure real-time communications providing software, cloud , core, and edge network infrastructure solutions to service providers and enterprises.

Sonus GENBAND edgewater NETWORKS

About Ribbon

Four Decades of Combined Leadership Experience in Real Time Communications

~ 2,300 Employees and Doing Business in 100+ countries

1,000+ Service Provider and Enterprise Customers Globally

#1 in VoIP Switching, #1 E-SBC, #2 CSP SBC, #1 in Media Gateways

800+ Patents Worldwide

Publicly Traded Company on NASDAQ

Leadership Ranking Source: IHS Research and ExactVentures 3Q-2018 Market share data (Ribbon includes GENBAND, Sonus, and Edgewater)

Where You Will Find Us

The World's Leading Tier One Service Providers

The Largest Banks, Airlines, Retailers and Manufacturers across the Globe

More than 350 U.S. Department of Defense Locations

Ribbon Protect

Big-data Analytics to Secure Communications Networks

Use Cases

Video, Mobility, RTC&C, Vo-Wifi, WebRTC

RTC Security

Toll Fraud

Continuous Monitoring

Threat Intel Sharing

Intelligent Operations

Analytics

Accelerate Investigations

Added context to investigations, visualization, multi sourced data collection, automation, drill down

Improve Operations

Consolidate RTC tools, NW Policy enforcement, active monitoring, troubleshooting, SOC/SIEM integration

Big Data

Protect

- High-speed data ingestion
- Hadoop
- Data Enrichment
- ML / Behavior Analytics
- Incident Management
- GPU Acceleration

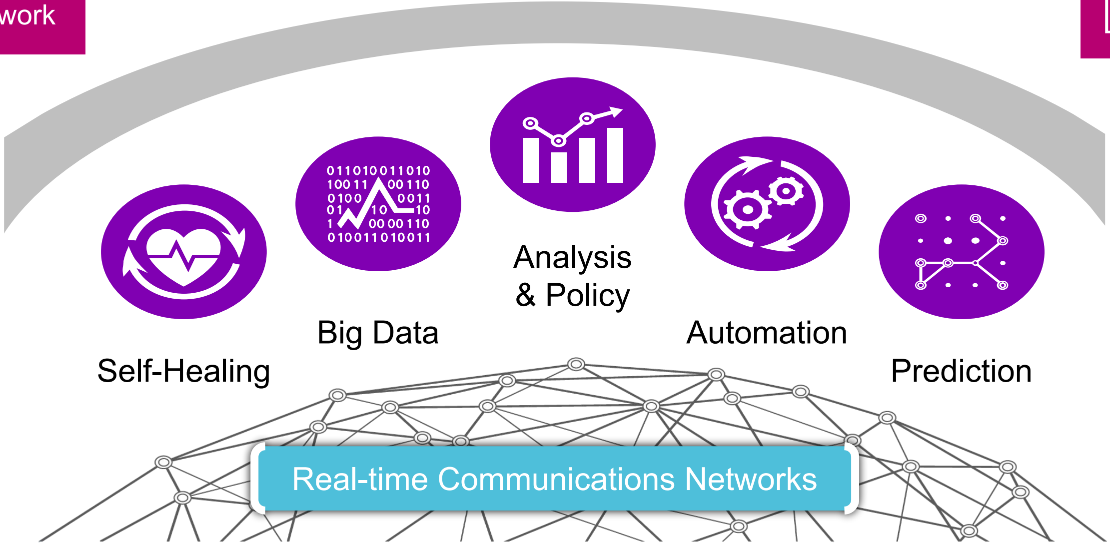
Sensors / Enforcers

Communications Network

AS, GSX9000, SBC, 3rd Party SBC, WRTC, C3, C20, Firewall, PSX, IP-PBX, G9

Goals: *Use Deep Learning to model calls*

- Anomaly detection
- Forecast utilization
- Behavioral user / network
- Call signatures



Modeling calls in a real-time Communication Network

Challenges

Network Complexity

Network behavior varies greatly between operators.

Machine learning models must be built and trained with operators data to capture the unique characteristics of their network.

Feature significance vary from operator to operator and may change over time

Data Dimensionality

Input sources contain high dimensional, text based data that results in large features sets

Metrics(KPI's) used for behaviors models can number from 10's to 1000's which presents significant resource challenges.

Analytics Scale

Call rates per sec
(in 10's of thousands)
pose challenges for real-time based modeling and detection

Billions of records per day for analytical processing

Security incidents and operational events can take significant time to detect

The Approach

Initial Key Focus Areas

Parameterize

Apply machine learning techniques to create features for call flows, user behavior and endpoint information

Model

Leverage deep learning to model typical or normative behavior such that anomalies can be readily identified and acted on

Operational

Forecasting and thresholding network KPI's
Identifying anomalous behaviors on network resources

Security

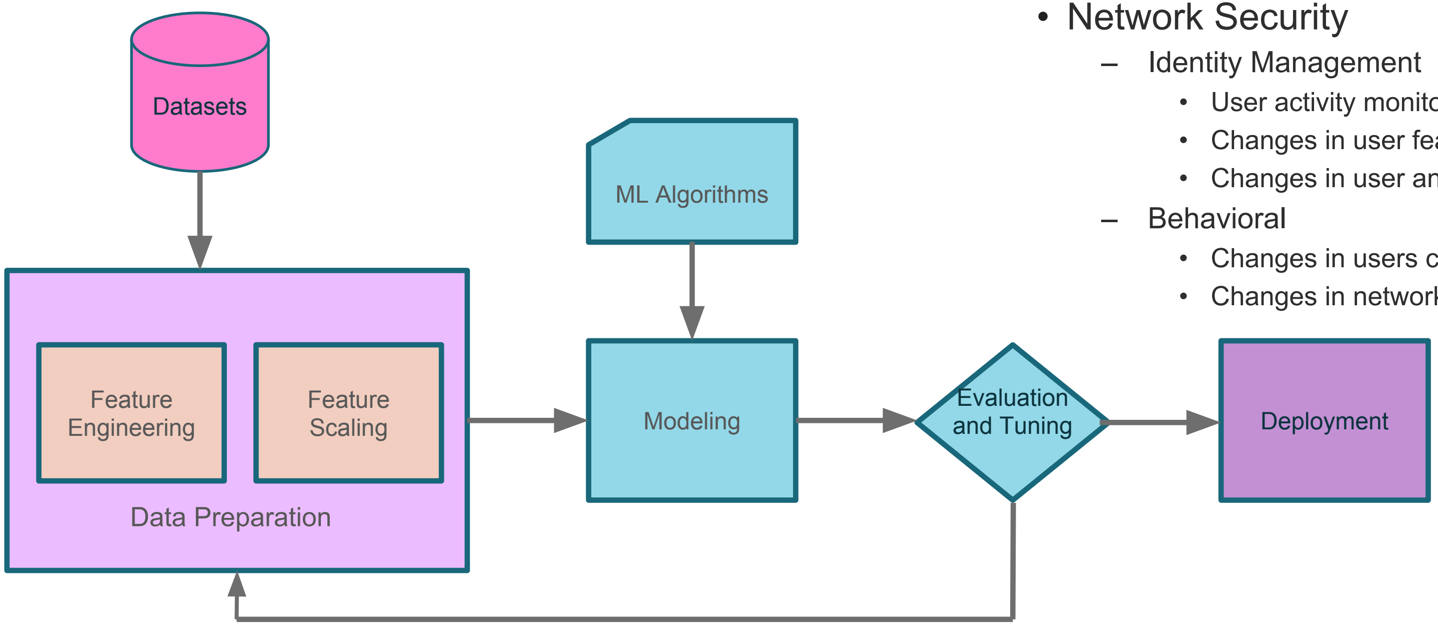
Behavioral modeling of subscribers usage and network calling patterns
Identifying security anomalies of subscribers actions

ribbon communications

SIP Call Signature

Hypothesis

Use Call signaling information to create a "signature"



Applications

- Service Assurance (Operational)
  - Understand types of devices on network
  - Onboarding new devices
  - Determining distribution of devices
- Network Security
  - Identity Management
    - User activity monitoring (think bank and credit card)
    - Changes in user features as compared to corpus
    - Changes in user and device relationships
  - Behavioral
    - Changes in users calling patterns
    - Changes in network usage

Unified Communications Data Sources

CDR – Call Detail Records

- Created at the beginning and end of calls (ATTEMPT, START, STOP)
- CSV format with 300+ columns
- Contains summary information about the calls (duration, quality, packets).
- Typically used for operator billing

Logs/pCap (SIP Messages)

- Unstructured text
- Much higher data volume than CDR
- Requires protocol parsing to parameterize
- Minimum of 4 messages per call

Challenge in building machine learning solution

Lack of labelled data

- Getting access to enough training data
- Diversity of device types

Scope of data attributes

- Device types, call types, device configurations/options, network modifications

ribbon communications

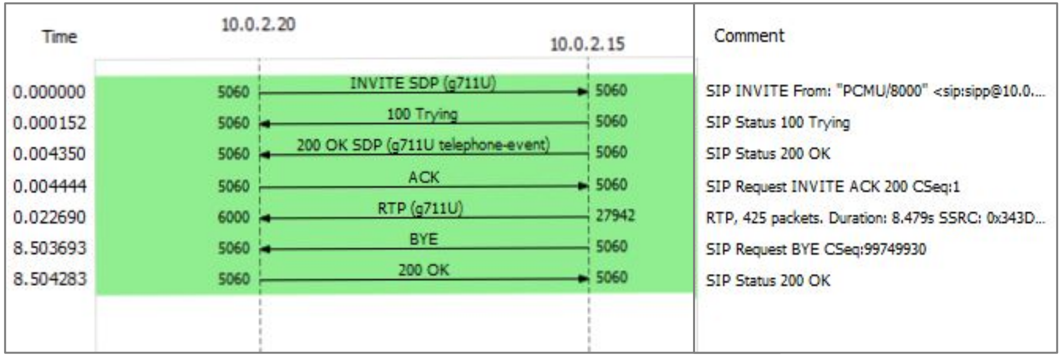
Session Initiated Protocol (SIP) Overview

```
INVITE sip:+17325551234@10.2.0.1:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.1:0;branch=z9hG4bK-14243-27817-0
From: +13155559999 <sip:+ 13155559999 @192.168.1.1:0>;tag=14243SIPpTag0027817
To: +17325551234 <sip:+17325551234@10.2.0.1:5060>
Call-ID: 387A9EFB@192.168.1.1
CSeq: 1 INVITE
Contact: sip:+ 13155559999 @192.168.1.1:0
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length:   137

v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.1
s=-
c=IN IP4 192.168.1.1
t=0 0
m=audio 6001 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

What is SIP

- Text based protocol
- Similar to HTTP
- “Soft” standard
  - *Syntax*
  - *Parameters*
  - *Extensibility*
- Lends to vendor specific implementations which we can leverage



SIP Message – Device features

Identify “what” is making a call

```
INVITE sip:+17325551234@10.2.0.1:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.1:0;branch=z9hG4bK-14243-27817-0
From: +13155559999 <sip:+ 13155559999 @192.168.1.1:0>;tag=14243SIPpTag0027817
To: +17325551234 <sip:+17325551234@10.2.0.1:5060>
Call-ID: 387A9EFB@192.168.1.1
CSeq: 1 INVITE
Contact: sip:+ 13155559999 @192.168.1.1:0
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length:   137

v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.1
s=-
c=IN IP4 192.168.1.1
t=0 0
m=audio 6001 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Header inclusion/exclusion
Format, parameters
Header Order
Syntax

SIP Message – User features

Identify “who” is making this call

```
INVITE sip:+17325551234@10.2.0.1:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.1:0;branch=z9hG4bK-14243-27817-0
From: +13155559999 <sip:+ 13155559999 @192.168.1.1:0>;tag=14243SIPpTag0027817
To: +17325551234 <sip:+17325551234@10.2.0.1:5060>
Call-ID: 387A9EFB@192.168.1.1
CSeq: 1 INVITE
Contact: sip:+ 13155559999 @192.168.1.1:0
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length:   137

v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.1
s=-
c=IN IP4 192.168.1.1
t=0 0
m=audio 6001 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

User identification
User parameters
Route (via)
IP information

SIP Message – Destination features

Identify "where" the call is going

```
INVITE sip:+17325551234@10.2.0.1:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.1:0;branch=z9hG4bK-14243-27817-0
From: +13155559999 <sip:+ 13155559999 @192.168.1.1:0>;tag=14243SIPpTag0027817
To: +17325551234 <sip:+17325551234@10.2.0.1:5060>
Call-ID: 387A9EFB@192.168.1.1
CSeq: 1 INVITE
Contact: sip:+ 13155559999 @192.168.1.1:0
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length:   137

v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.1
s=-
c=IN IP4 192.168.1.1
t=0 0
m=audio 6001 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Destination information
Type of call
Media information

SIP Message – Call features

Identify details of this call

```
INVITE sip:+17325551234@10.2.0.1:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.1:0;branch=z9hG4bK-14243-27817-0
From: +13155559999 <sip:+ 13155559999 @192.168.1.1:0>;tag=14243SIPpTag0027817
To: +17325551234 <sip:+17325551234@10.2.0.1:5060>
Call-ID: 687A9EFB@192.168.1.1
CSeq: 1 INVITE
Contact: sip:+ 13155559999 @192.168.1.1:0
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length:   137

v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.1
s=-
c=IN IP4 192.168.1.1
t=0 0
m=audio 6001 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Identify of specific call

Calling,Called

Call idenfication attributes

  callId, Tags, Routing

Type of call

Statistics (duration, etc)

ribbon communications

Creating Machine Learning Features
Data Preparation

Example of a few techniques used to create features from SIP messages:

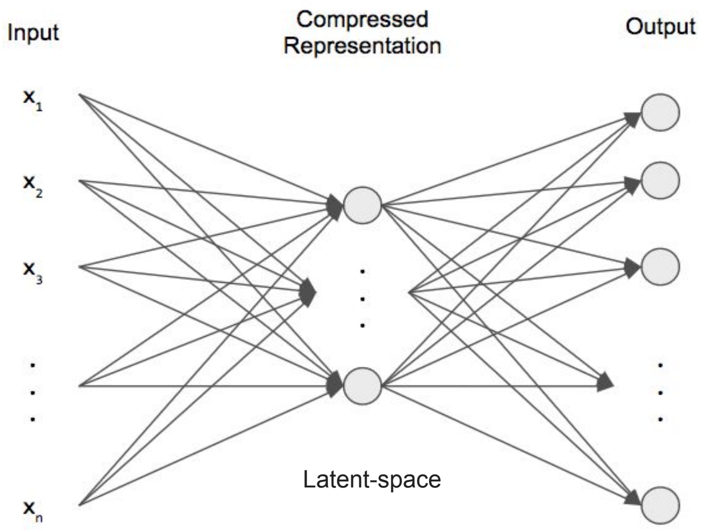
- **Header Presence** – for each header in message identify number of occurrences
- **Header Sequence** – identifies the sequence or order of a header in the message
- **Header Syntax** – the original message syntax for the header name (upper/lower)
- **Masks** – creates a format mask for implementing specific SIP parameters.
  - Typically helpful to identify a device specific implementation
- **Where:**
  - N – numeric
  - U – upper case
  - L – lower case
  - S – space
  - X – special character
  - Z – other
- **Example**: Encoding tag value contained in from header
  - From: …;tag=14243SIPpTag0027817 -> NNNNNUUULULLNNNNNNN

Choosing a Machine Learning Model

What can we do with the data we have ?

- Limited to an Unsupervised Learning model
  - Looked at various clustering models
  - Neural networks generative models promising
    - Autoencoder seems to fit our problem
    - - *Tested with several autoencoder configurations*
    - - *Variational autoencoder provided best results*

Autoencoders are a specific type of feedforward neural networks where the input is the same as the output. They compress the input into a lower-dimensional code and then reconstruct the output from this representation. The code is a compact “summary” or “compression” of the input, also called the latent-space representation.



- Use SIP featured to train multiple autoencoders
  - Device, User, Destination, Call models
    - Use latent-space(compressed) as a digital signature for each feature

Implementing an Autoencoder

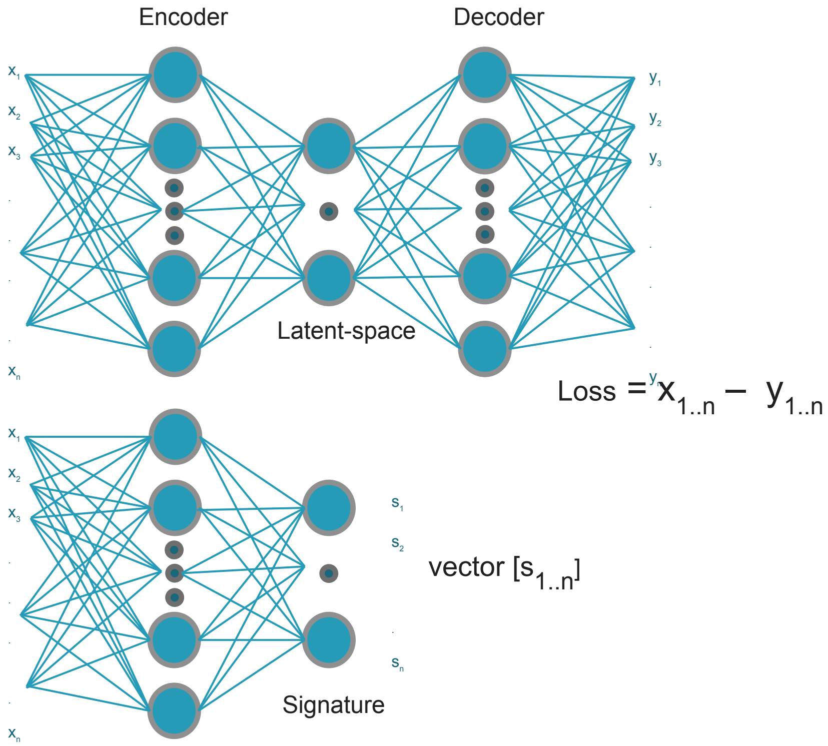
Creating a "signature"

Training phase

- Autoencoder minimizes loss between input features and output features of the training data
- Latent-space layer compresses the learned information from the input features

Operational phase

- Trained model uses only *Encoder* portion of network
- Latent-space vector becomes the 'signature'



ribbon communications

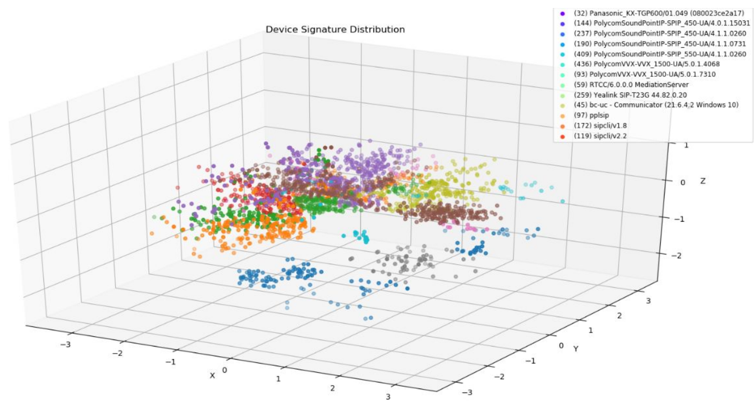
Service Assurance

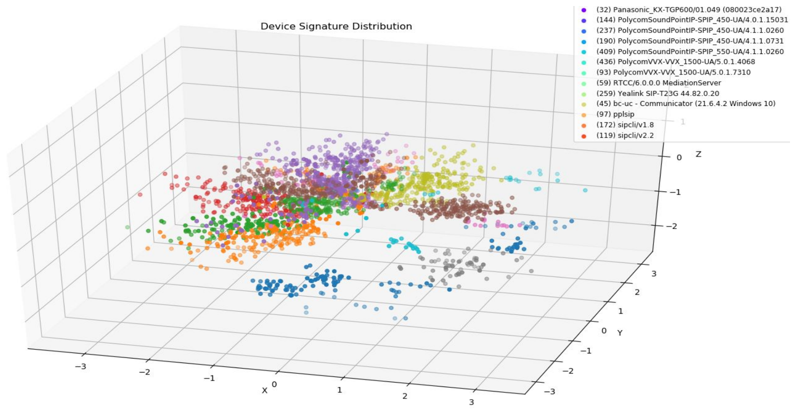
Using AI to enhance network operations

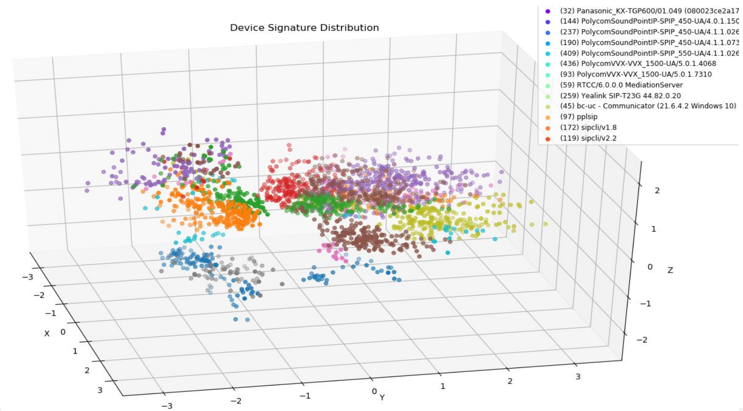
- **Provide visibility into operators network**
  - With *Device* features:
    - Mapping devices in network
      - Metadata provides visibility into device attributes
    - Determine density of device types
    - Notification of new device types in network
  - With *User* and *Destination* features:
    - Identify call flow patterns
- **Operational actions**
  - Onboarding new device types
    - Identify network interoperability requirements
  - Network Resource Management
    - Capacity planning and forecasting

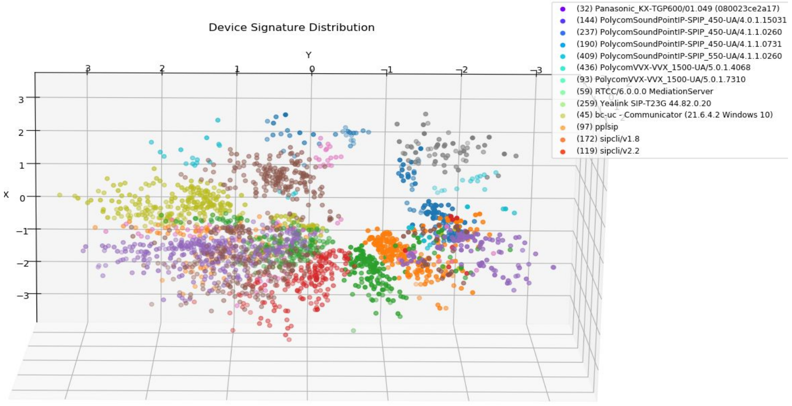
Device Autoencoder

Signature visualization

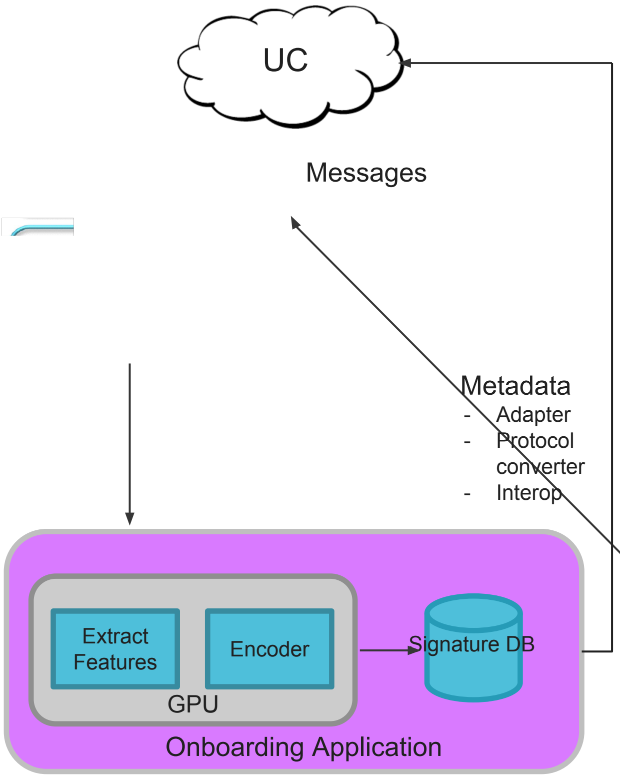


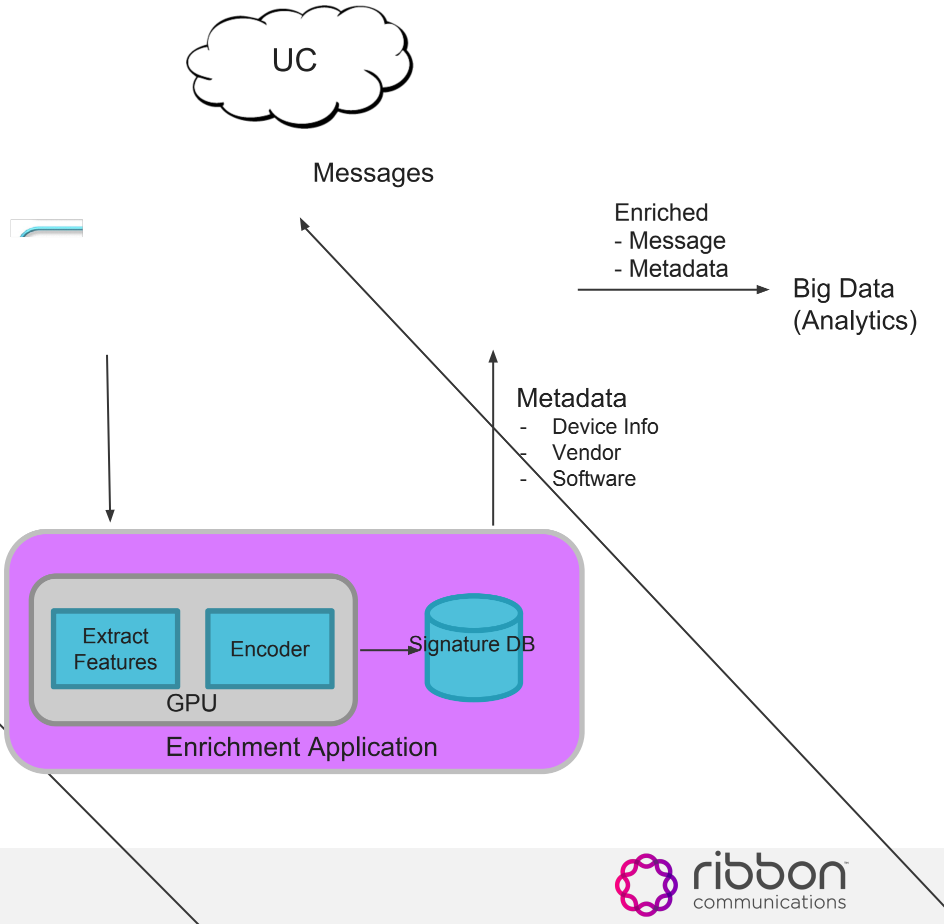






Service Assurance
Application Examples





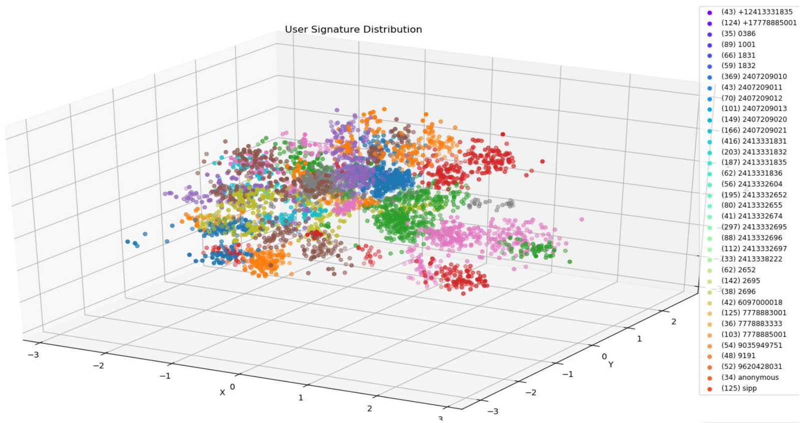
Identity Management

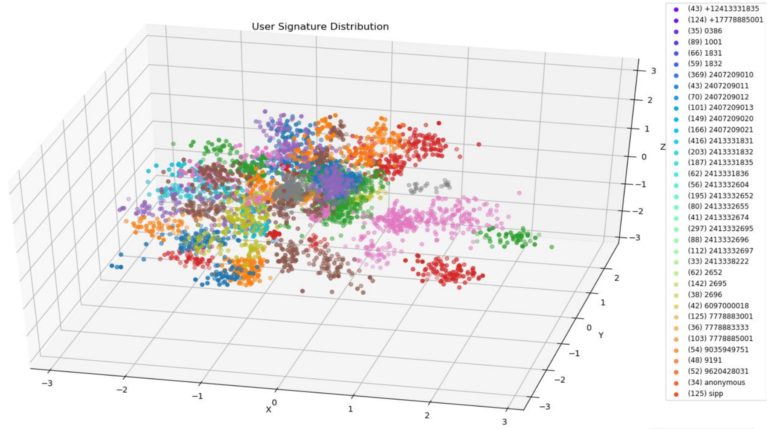
Using AI to protect network and users

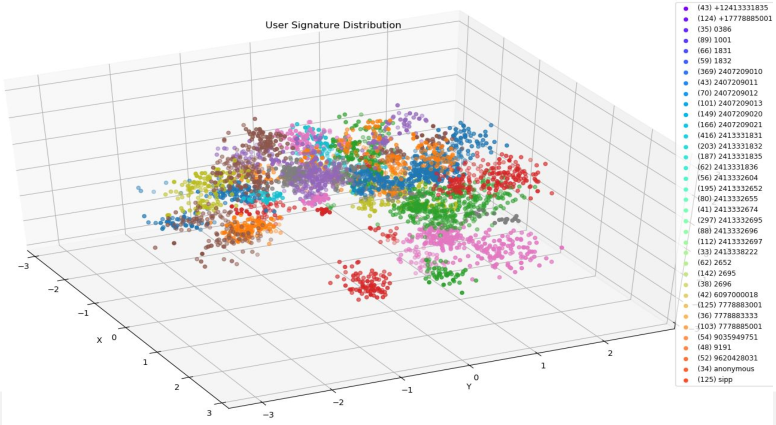
- **Provide insight to endpoints and users**
  - With *Device* features:
    - Identify malicious or misbehaving devices
    - Reporting new or unknown types of device
  - With *Device* with *User* features:
    - Verification of user and device signatures
      - *Location (geo)*
      - *Device type with this user*
    - Detecting concurrent user instances
- **Security actions**
  - Block malicious devices and users
  - Identify security vulnerabilities in network devices
  - Feed anomalies into fraud applications
  - Generate incidents to SIEM

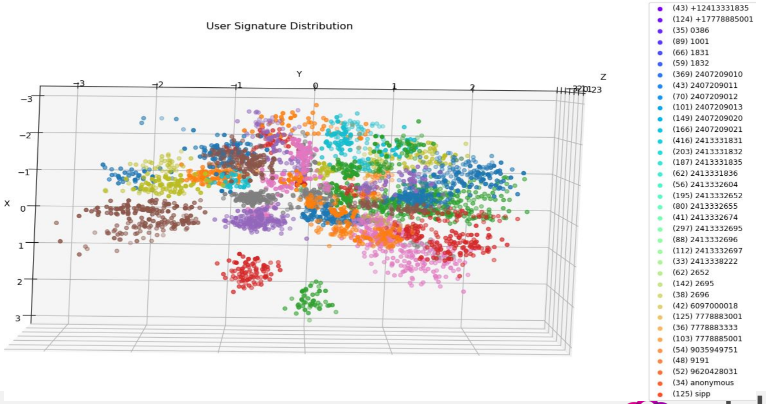
User Autoencoder

Signature visualization





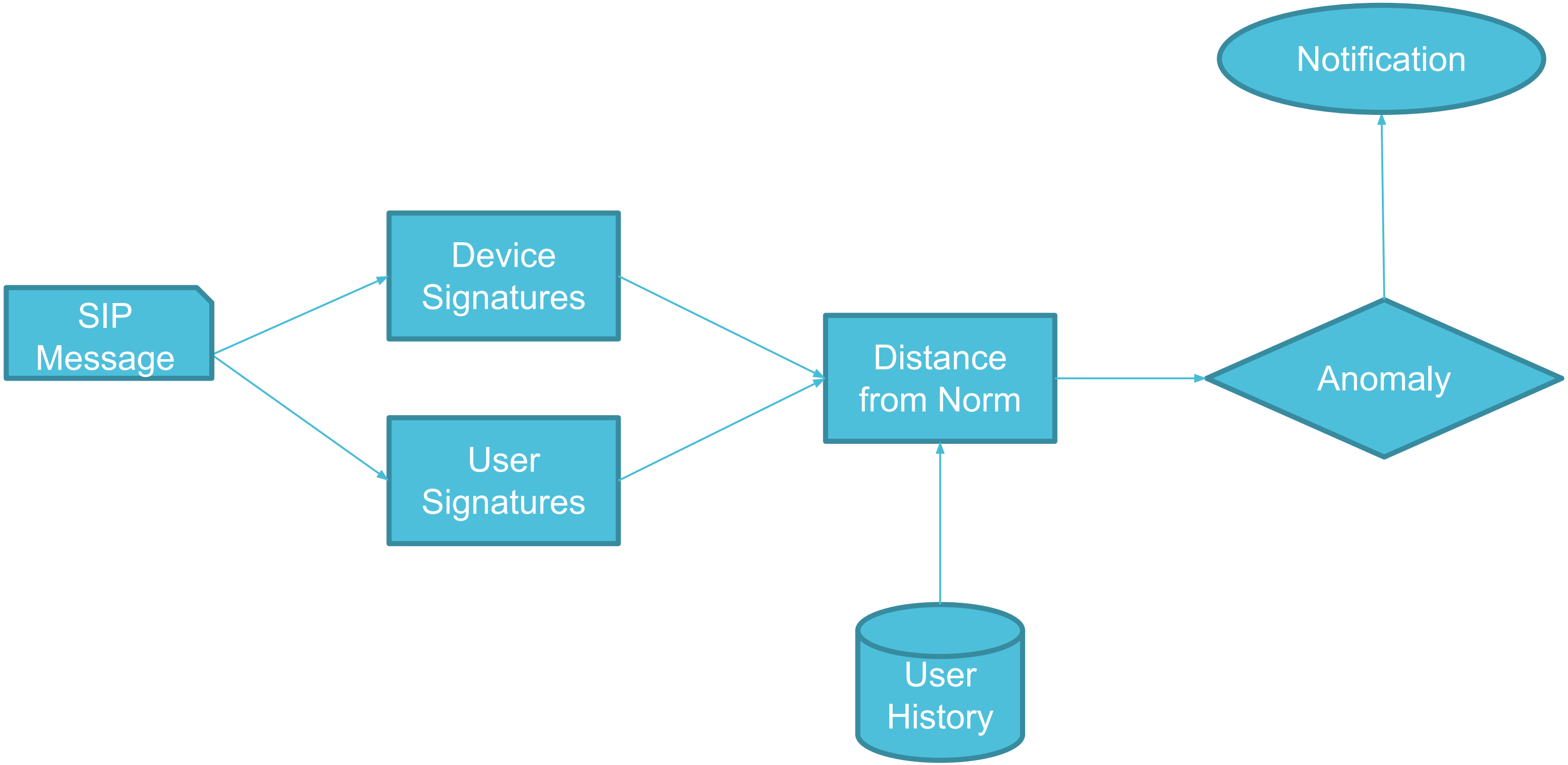




ribbon communications

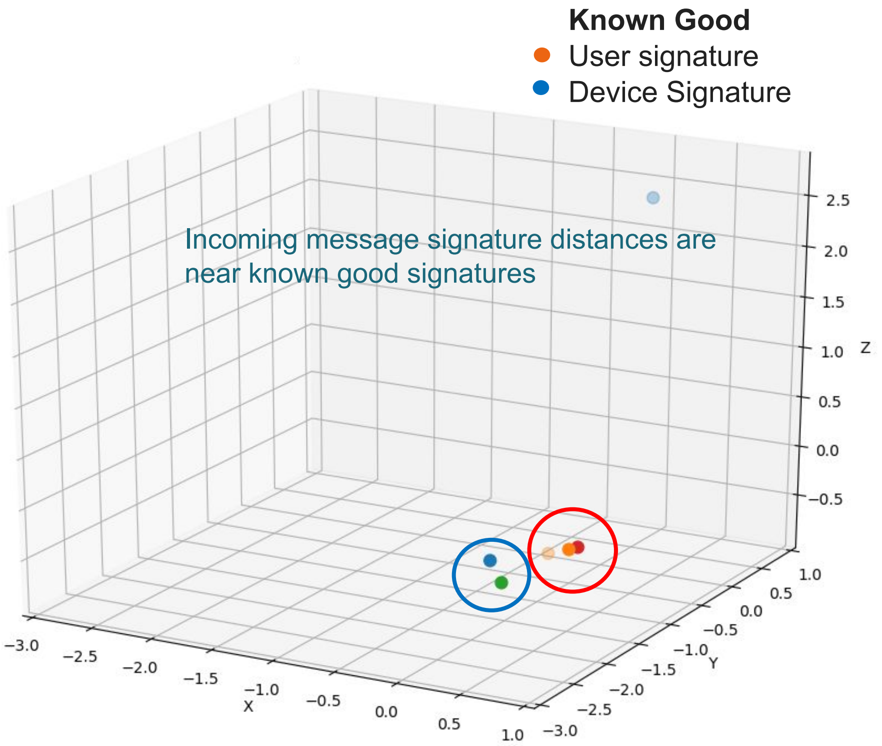
Detecting Anomalies

Combining signatures for more advanced analysis



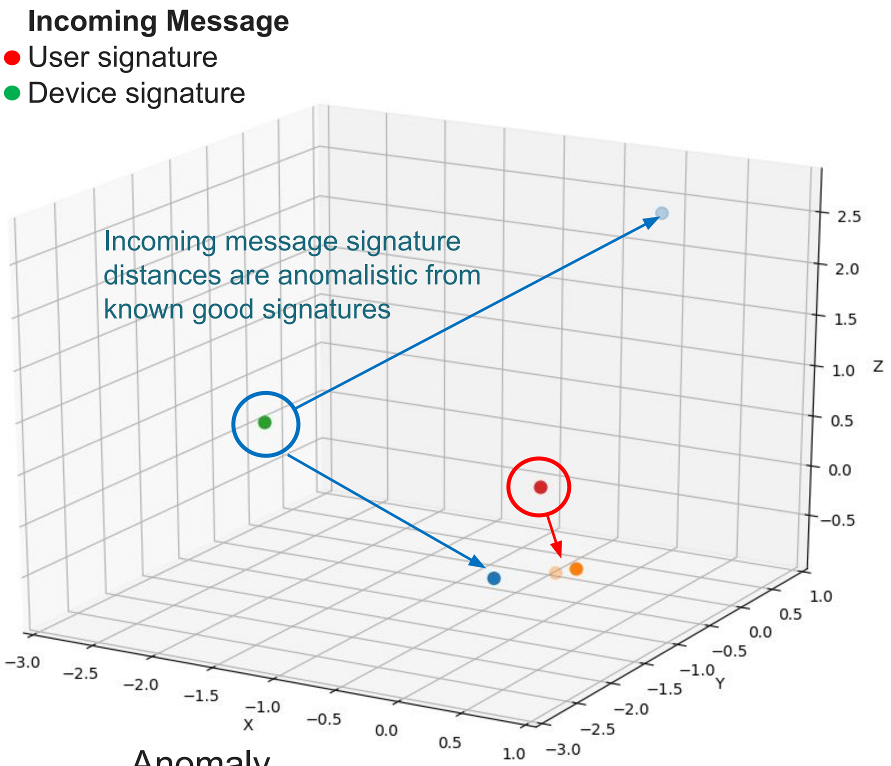
Detecting Anomalies - example

Combining device and user signatures



Normal

Minimum distance [0.2661066981234524, 0.08135181163868711]

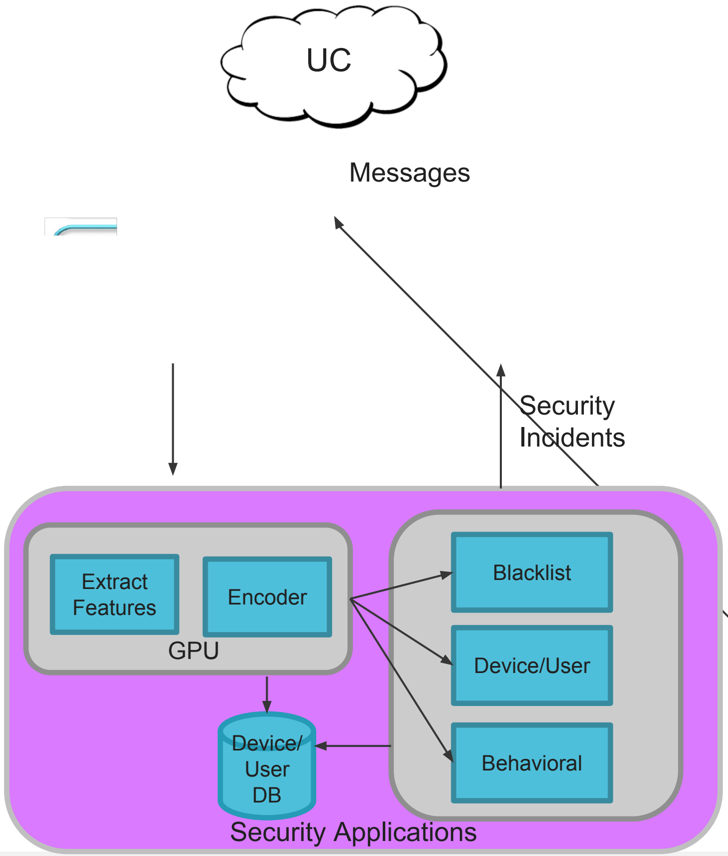


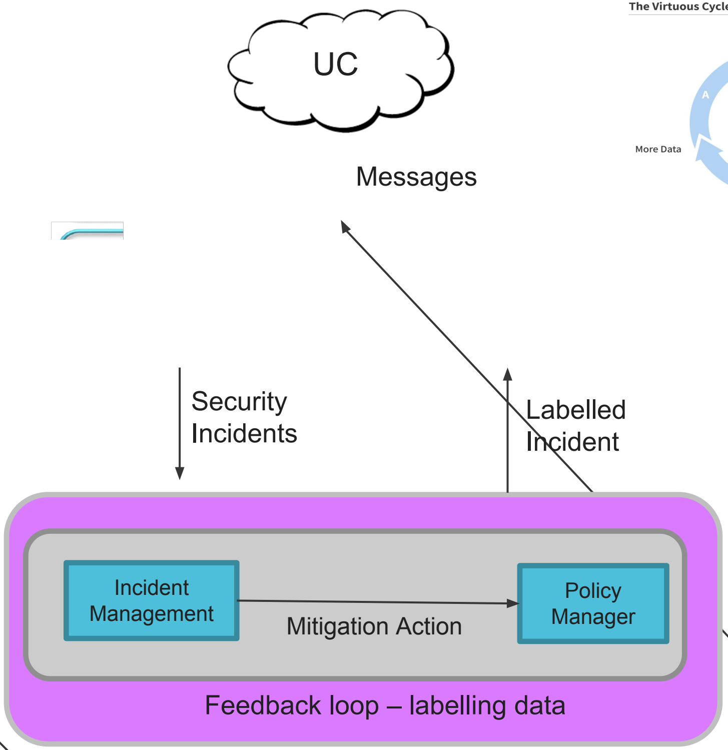
Anomaly

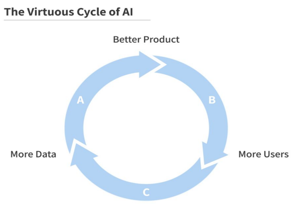
Minimum distance [3.211263703324535, 0.9243276898181685]

Identity Management

Application Examples



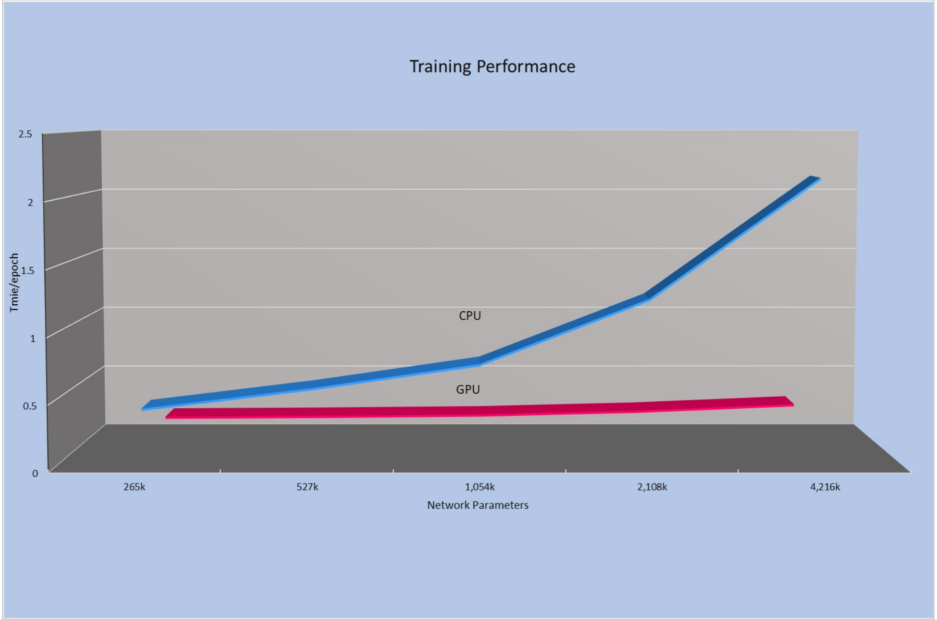


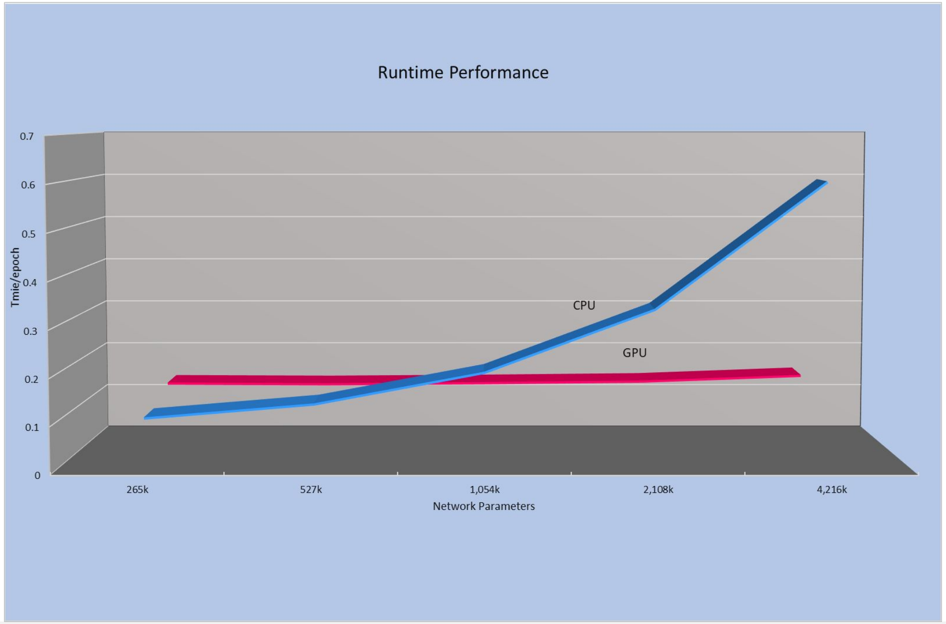


Hypothesis to Deployment

Performance demands require GPU's

- Scaling to production volume is a significant challenge
  - 10's of thousand calls/sec, 10's of MB of data/sec
  - Ingestion, extraction and predicting/encoding are all bottlenecks
  - AI model complexity increase processing demands

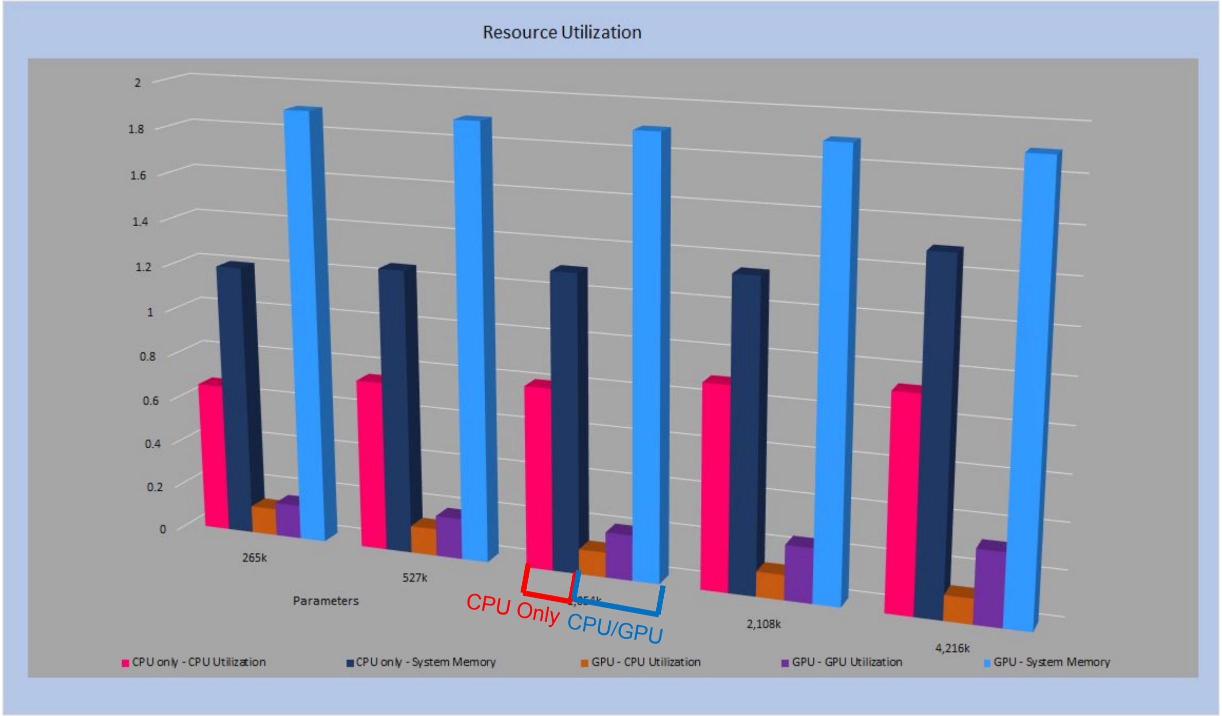




Maximizing System Resources

Choosing the right tool for the job

- How we split AI pipeline
  - *Ingestion optimization and enrichment through distributed CPU nodes*
  - *Data Preparation and filtering as a common CPU service*
  - *Model Predictions/Encoding through GPU*
  - *Results processed by CPU based applications*
- Nothing comes for free
  - *Data movement becomes new bottleneck*
  - *Using GPU, CPU memory consumption increases*

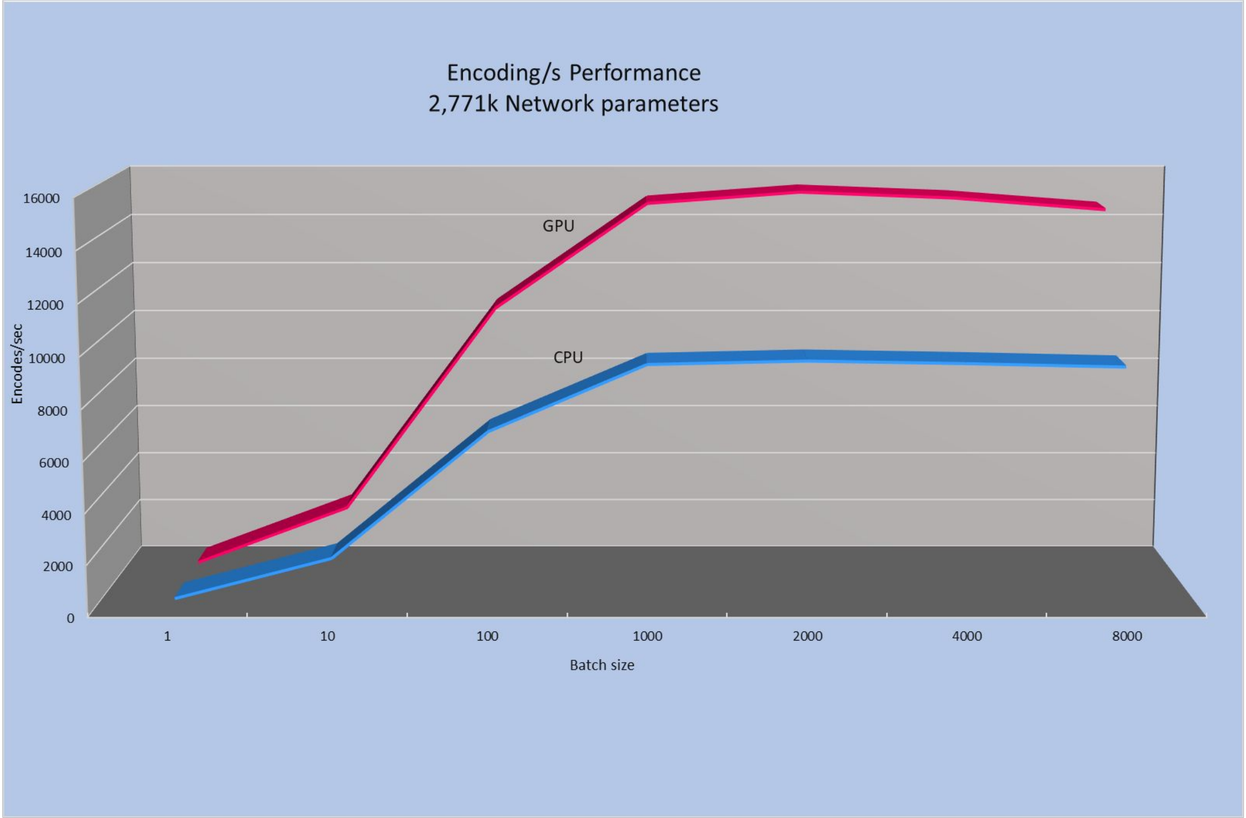


Actual performance impact using a GPU

Increasing the volume -"Turn it up to 11"

- Hardware
  - *I7-8700K 3.7G 6 Core*
  - *32 GB Memory*
  - *1T SSD HD*
  - *Nvidia GTX-1080*
- *Software*
  - *Python 3.6.6*
  - *Tensorflow 1.11.0*
  - *Keras*

- *AI pipeline performance*
  - *Model with 2.7M network parameters*
  - *Varied encode batch size from 1-8000*
- *Results*
  - *Optimal batch size – 1000*
  - *GPU 15,366 encodes/sec*
  - *CPU 9,433 encodes/sec*



Summary

- **Ribbon is using AI to create new applications for Service Providers**
  - Initial focus on Service Assurance and Identify Management
    - Signatures for call flows
  - AI is enabling innovative solutions and advanced analytic capabilities
    - Anomaly detection
    - Forecasting
  - Building knowledge through system deployment
    - Labeling data
- **Ribbon & NVIDIA**
  - NVIDIA GPU's enhance Ribbon Protect to meet the scaling requirements of our customers and applications
  - NVIDIA resources (tools and libraries) address many or our development hurdles
    - Lets Ribbon focus on value added applications
    - NVIDIA Kubernetes distribution & NVIDIA Container runtime – easy integration into Ribbon Protect
    - RAPIDS – researching how RAPIDS can to improve our AI pipeline processing
  - NVIDIA has been a great partner in teaching, listening, and supporting Ribbon in it's path down AI

Thank You

ribbon™
communications