Defeating the Curse of "First Time" Events

Ignacio Bermudez Corrales | Security Data Scientist @SplunkUBA
icorrales@splunk.com

May 2018 | Version 1.0

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

splunk> .conf18

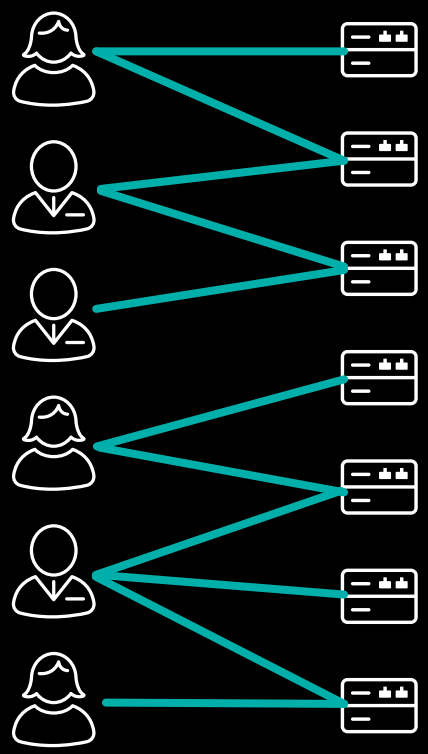
What are “First Time Events”

And why we should care about them

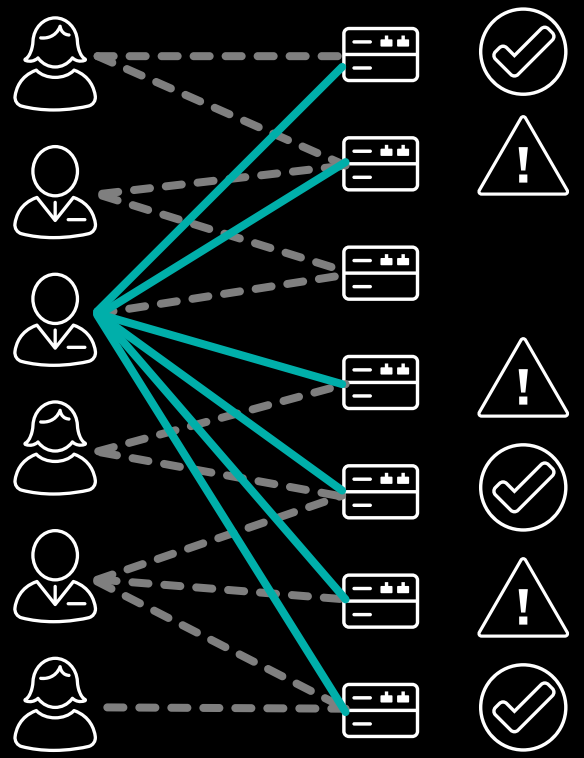
Machine Accesses

Do all first time accesses look suspicious?

Historical Accesses



First Time Accesses



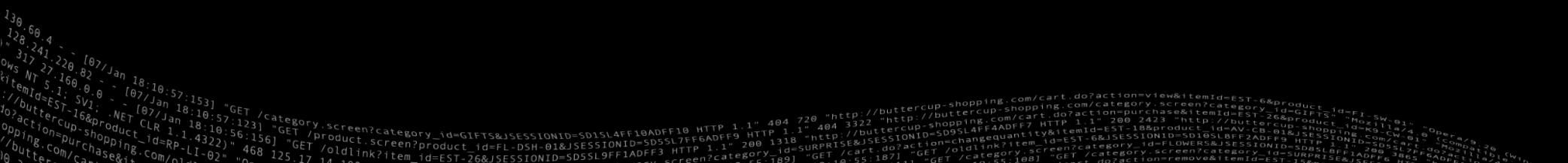
Normal Operations?

Suspicious Accesses?

Other Examples of First Time Events in Real Life

Some are normal, some can be risky

- Cloud Storage File-sharing
  - Employee is planning to leave the company and downloads some unprotected sensitive files
  - A newly made presentation is shared for review among some group of employees
- Private Code Repositories
  - An account is compromised and cloned projects with intellectual property
  - Cloned project made by a colleague that I'm planning to collaborate on
- Badge Accesses
  - A worker enters the office servers room during off work hours
  - Same worker enters some other campus' meeting room during a business trip



A Needle in a Haystack

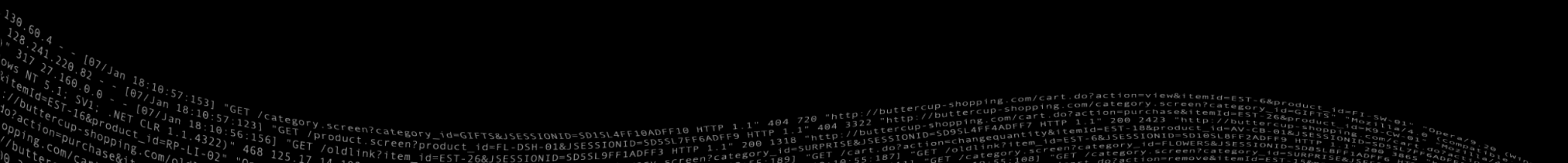
The risk of first time events

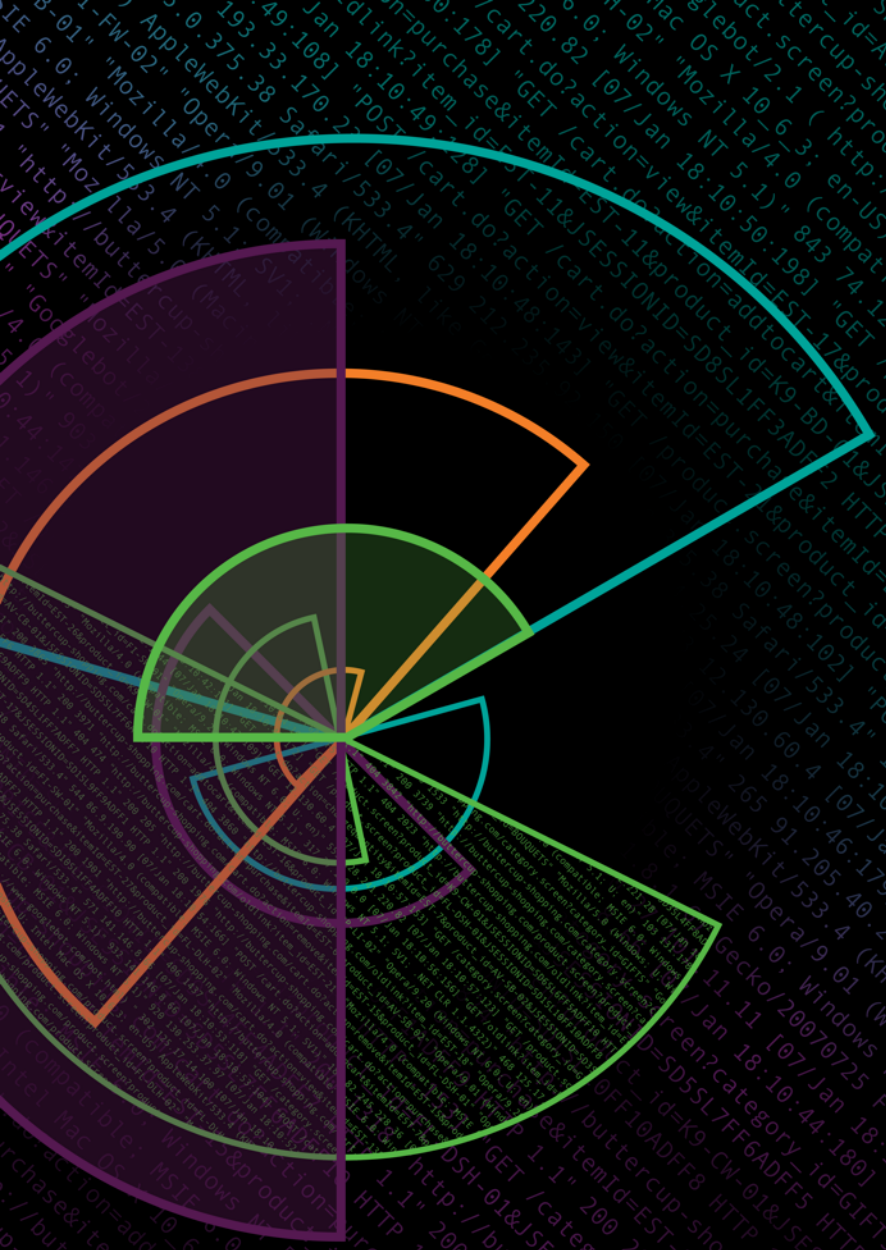
- Normal operations trigger a “sea” of first time events
- Collecting evidence for suspicious activity may take too long
- Suspicious activity can happen only once

Solution?

First time event handling

Conservative	Relaxed	Rules	Machine Learning
All first time events are interesting and worth investigating	First time events are not interesting	Low coverage, high precision	Finds the subtle, but learns exclusively from evidence



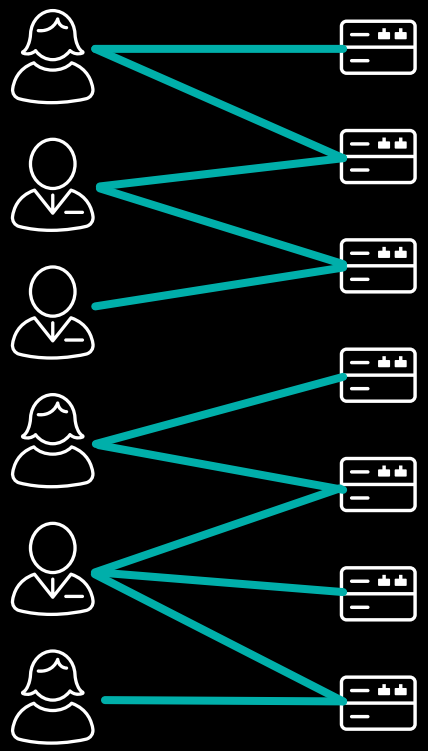


Let's Play with Numbers

Foundations

Representation of the Problem

So a machine learning algorithm can learn



Analogous Representations

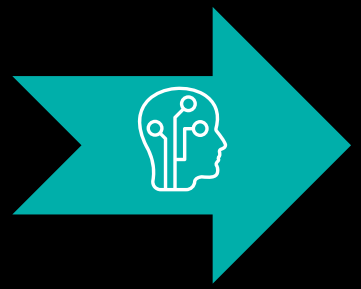
	1	1					
		1	1				
			1				
				1	1		
					1	1	1
							1

Machine Learning Generalizes

Filling the gaps

Historical Data - Evidence

	1	1					
		1	1				
			1				
				1	1		
					1	1	1
							1



ML Reconstruction of Data

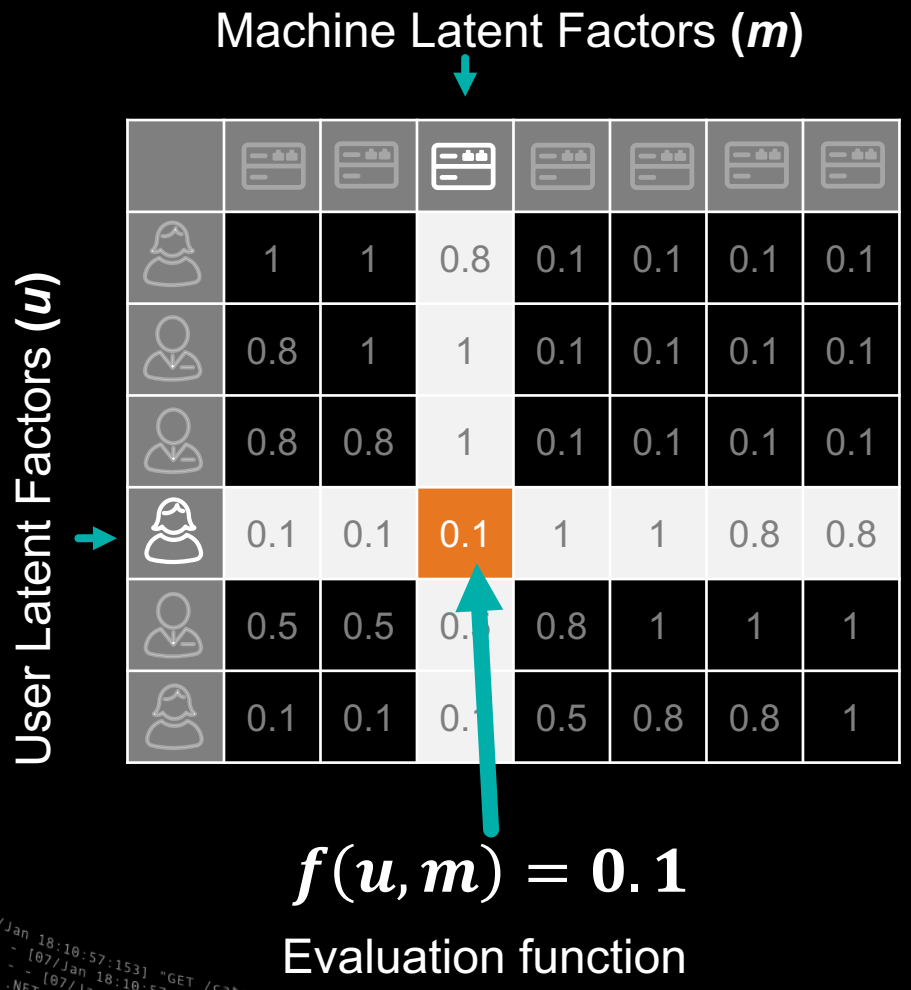
	1	1	0.8	0.1	0.1	0.1	0.1
	0.8	1	1	0.1	0.1	0.1	0.1
	0.8	0.8	1	0.1	0.1	0.1	0.1
	0.1	0.1	0.1	1	1	0.8	0.8
	0.5	0.5	0.5	0.8	1	1	1
	0.1	0.1	0.1	0.5	0.8	0.8	1

Suspicious — Benign

ML generalization capabilities allow Predictions of First Time Events

How Data is Reconstructed

Latent factor models



Latent factors (***u, m***) are real valued vectors which encode behavior and properties, that evaluated with a function allow reconstruction of the evidence.

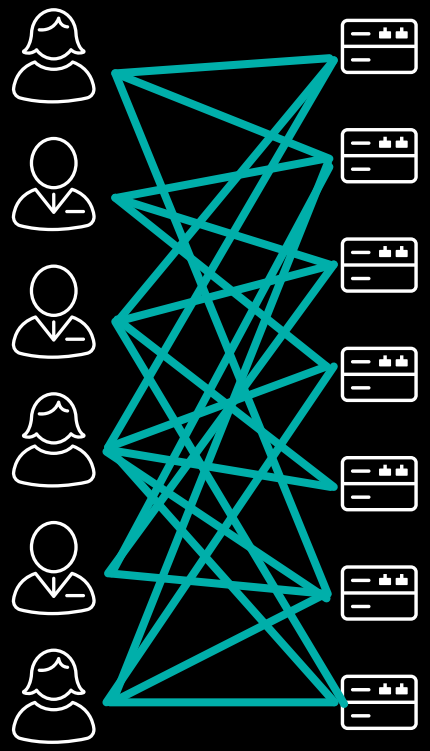
Netflix's recommendation system

Latent factors may be indicators of
Of age, income, gender of watchers;
And for movies the genre, duration, release date

Taming ML with Rules

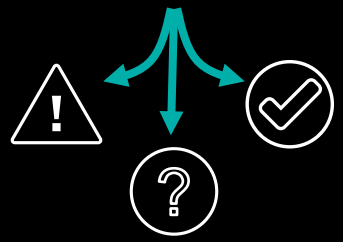
Passing down knowledge to machine learning algorithm

Synthetic Generated Events

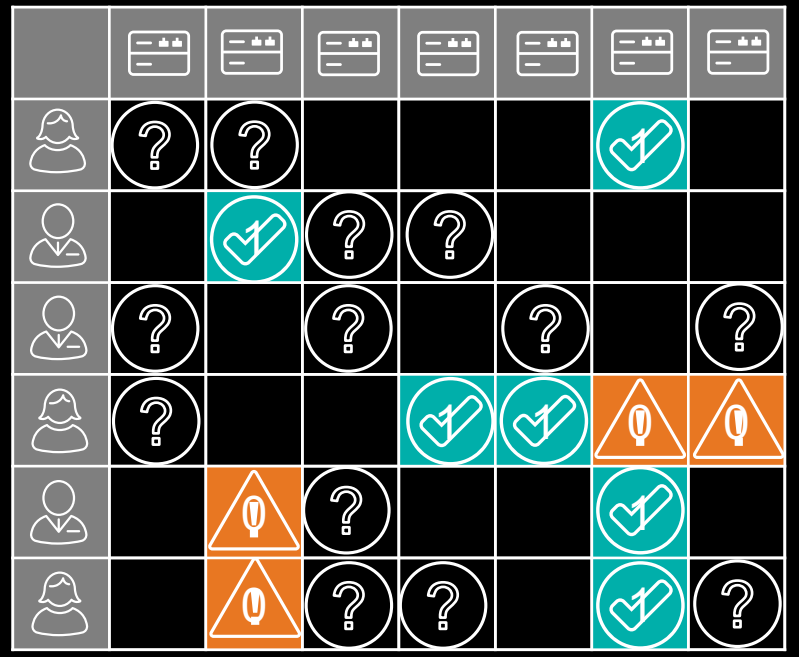


Rules

if(event == condition)

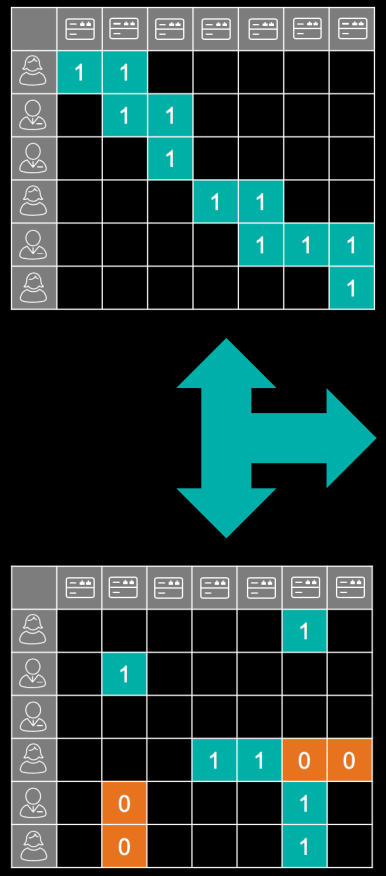


Coverage of Rules



Connecting Dots

Tamed machine learning embeds knowledge from rules and evidence



Historical Data + Rules

	1	1				1	
		1	1				
			1				
				1	1	0	0
		0			1	1	1
		0				1	1

ML Reconstruction of Data

	1	1	0.8	0.1	0.5	0.8	0.5
	0.8	1	1	0.1	0.5	0.5	0.5
	0.5	0.5	1	0.1	0.1	0.1	0.1
	0.1	0.1	0.5	1	1	0.1	0.1
	0.1	0.1	0.5	0.8	1	1	1
	0.1	0.1	0.5	0.5	0.8	0.8	1

Connecting Dots

Richer predictive model can describe more complexities

Before

	1	1	0.8	0.1	0.1	0.1	0.1
	0.8	1	1	0.1	0.1	0.1	0.1
	0.8	0.8	1	0.1	0.1	0.1	0.1
	0.1	0.1	0.1	1	1	0.8	0.8
	0.5	0.5	0.5	0.8	1	1	1
	0.1	0.1	0.1	0.5	0.8	0.8	1

Just evidence

After

	1	1	0.8	0.1	0.5	0.8	0.5
	0.8	1	1	0.1	0.5	0.5	0.5
	0.5	0.5	1	0.1	0.1	0.1	0.1
	0.1	0.1	0.5	1	1	0.1	0.1
	0.1	0.1	0.5	0.8	1	1	1
	0.1	0.1	0.5	0.5	0.8	0.8	1

Rules + Evidence

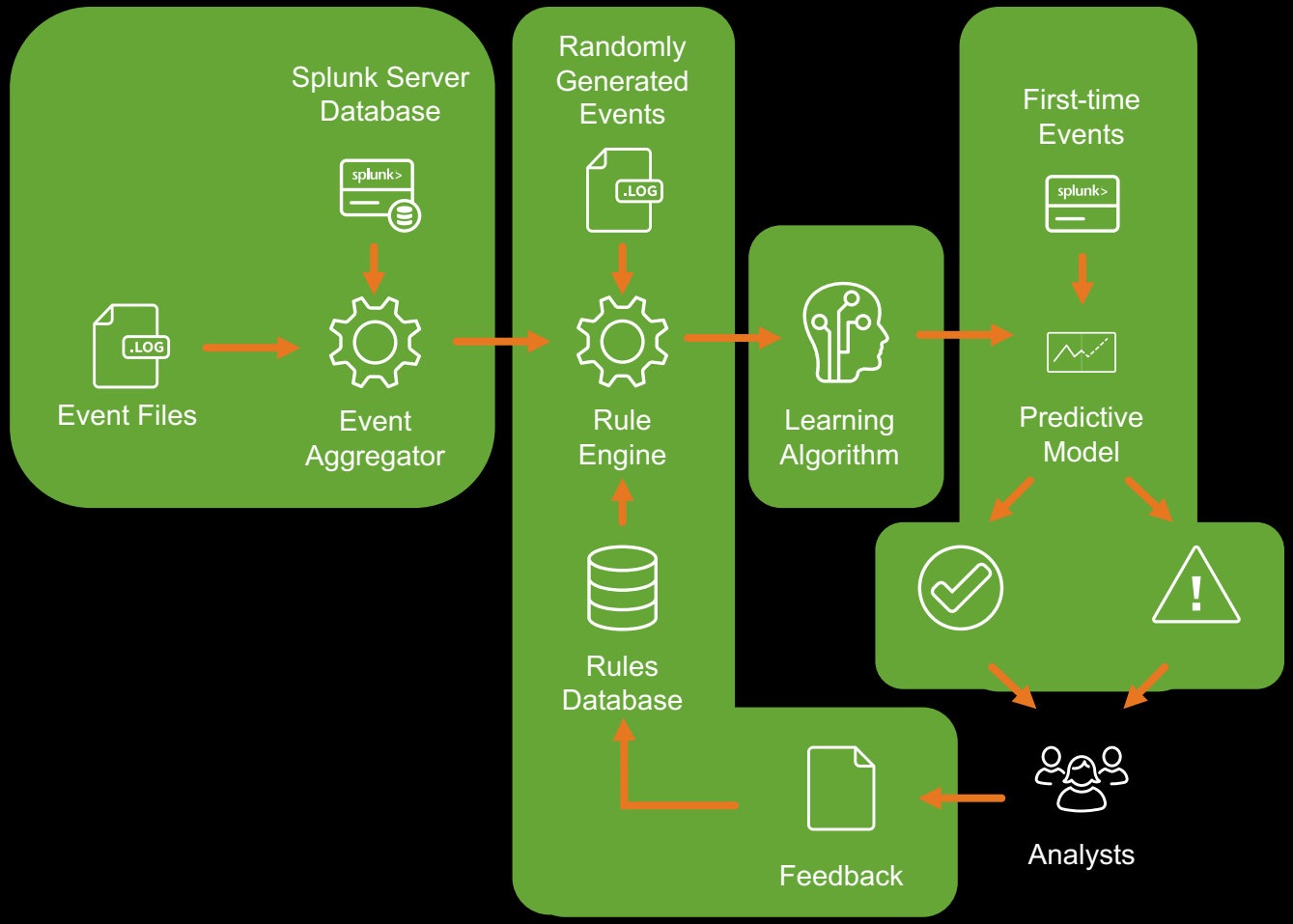
splunk> .conf18

Implementation

Building suspicious first event detector

Architecture

4 steps

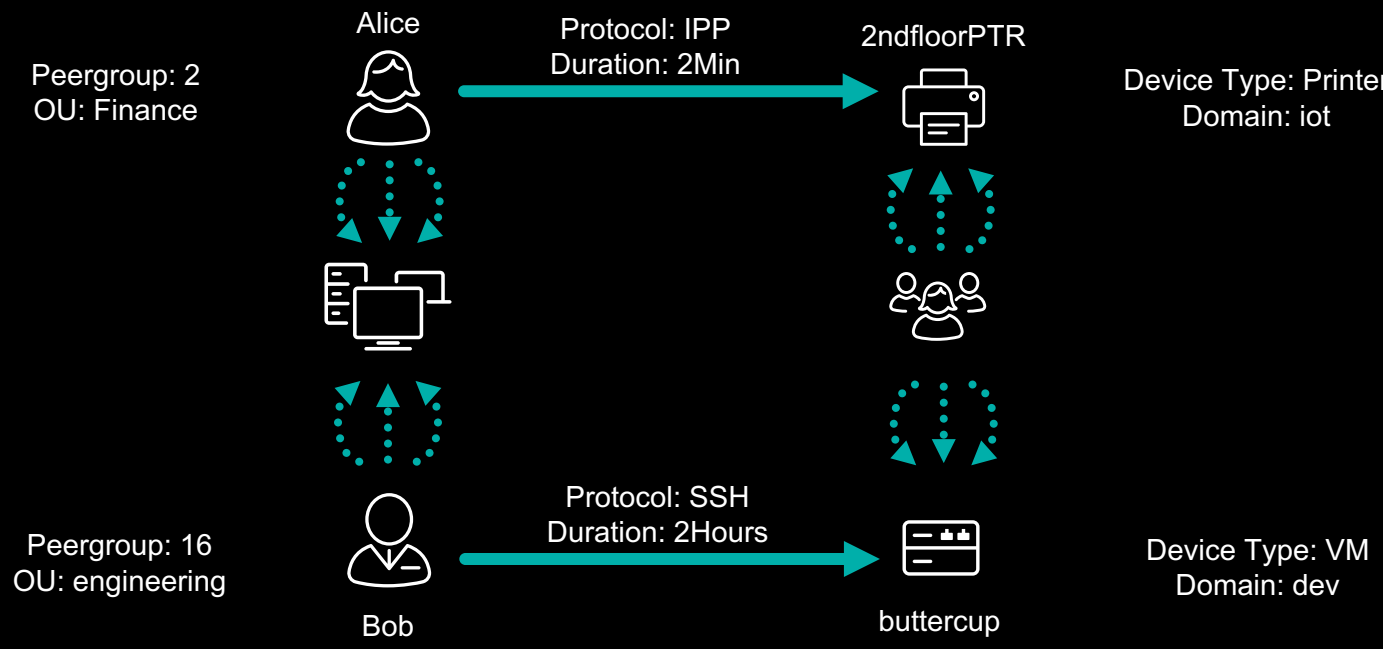


Example Case

Two employees access two devices

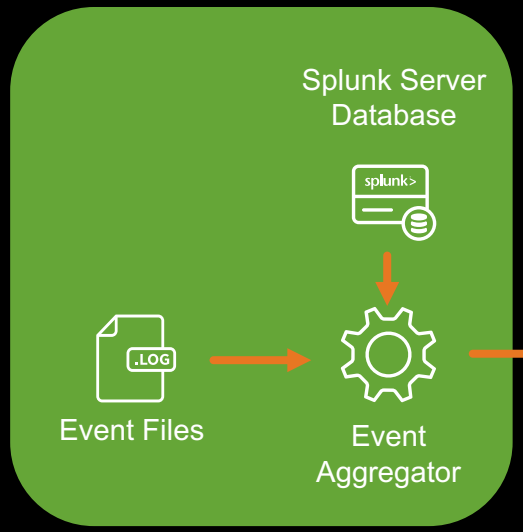
User Properties

Device Properties



Event Processing

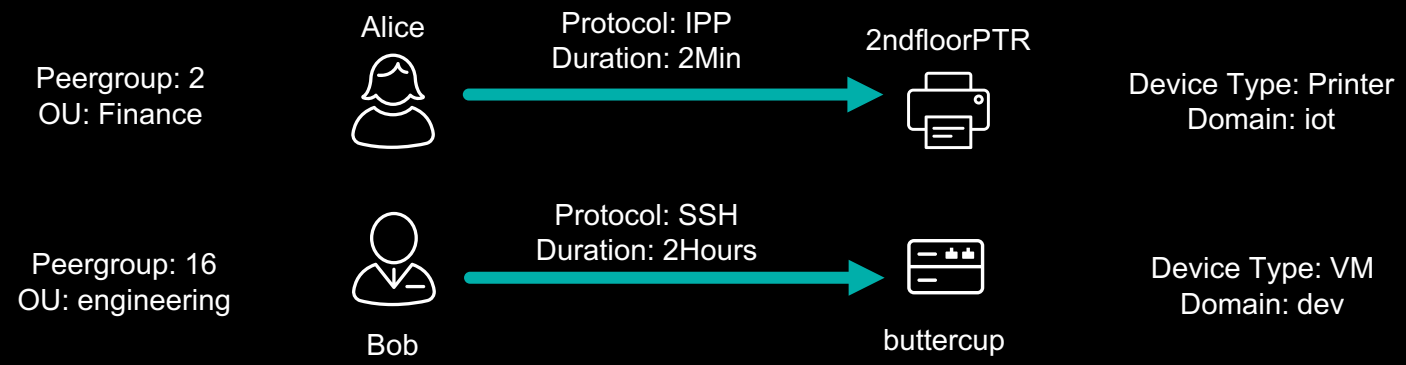
Step 1: Build a summary of information for the rule engine



Generates a summary of events
for the rule engine to make decisions

Event Processing

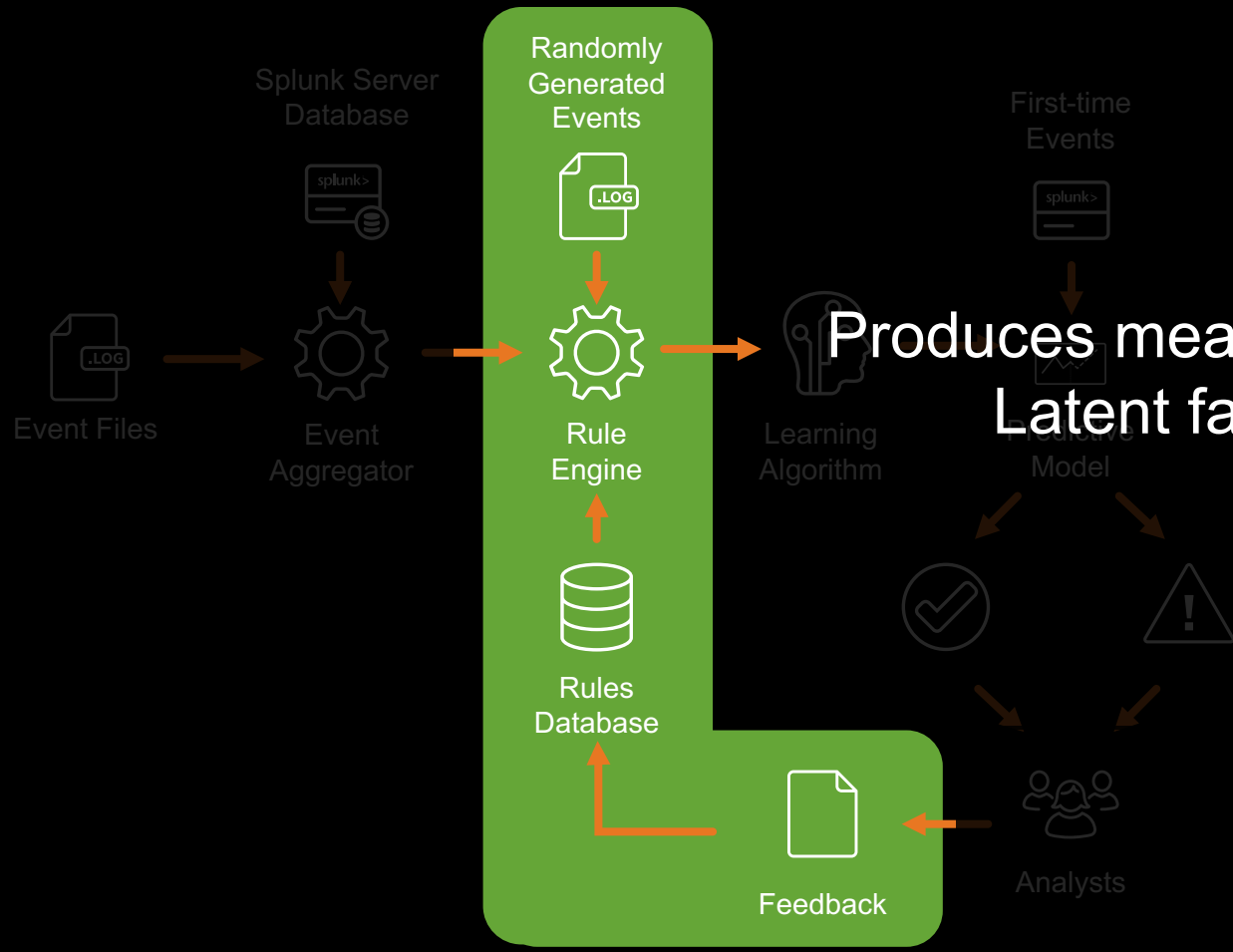
Step 1: Converting single event into an annotated graph with properties



Link ID	User Properties	User Name	Events Summary	Device Name	Device Properties
1	Peergroup: 1 OU: Engineering	Bob	SSH, 2h	Buttercup	Type: VM Domain: dev
2	Peergroup: 10 OU: Finance	Alice	IPP, 2m	2ndfloorPRT	Type: Printer Domain: iot

Rule Engine

Step 2: Convert event summaries into numerical signals

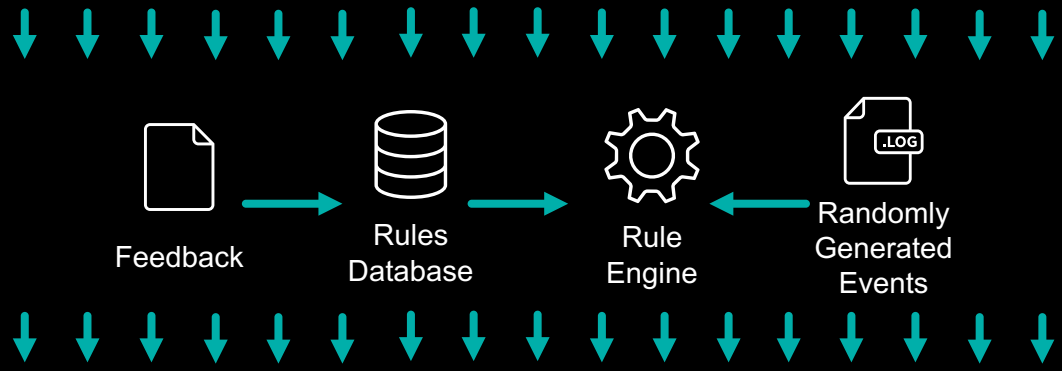


Produces meaningful training data for Latent factor ML algorithm

Rule Engine

Step 2: Converting links into numerical signals

Link ID	User Properties	User Name	Events Summary	Device Name	Device Properties
1	Peergroup: 1 OU: Engineering	Bob	SSH duration 2h	Buttercup	Type: VM Domain: dev
2	Peergroup: 10 OU: Finance	Alice	IPP	2ndfloorPRT	Type: Printer Domain: iot



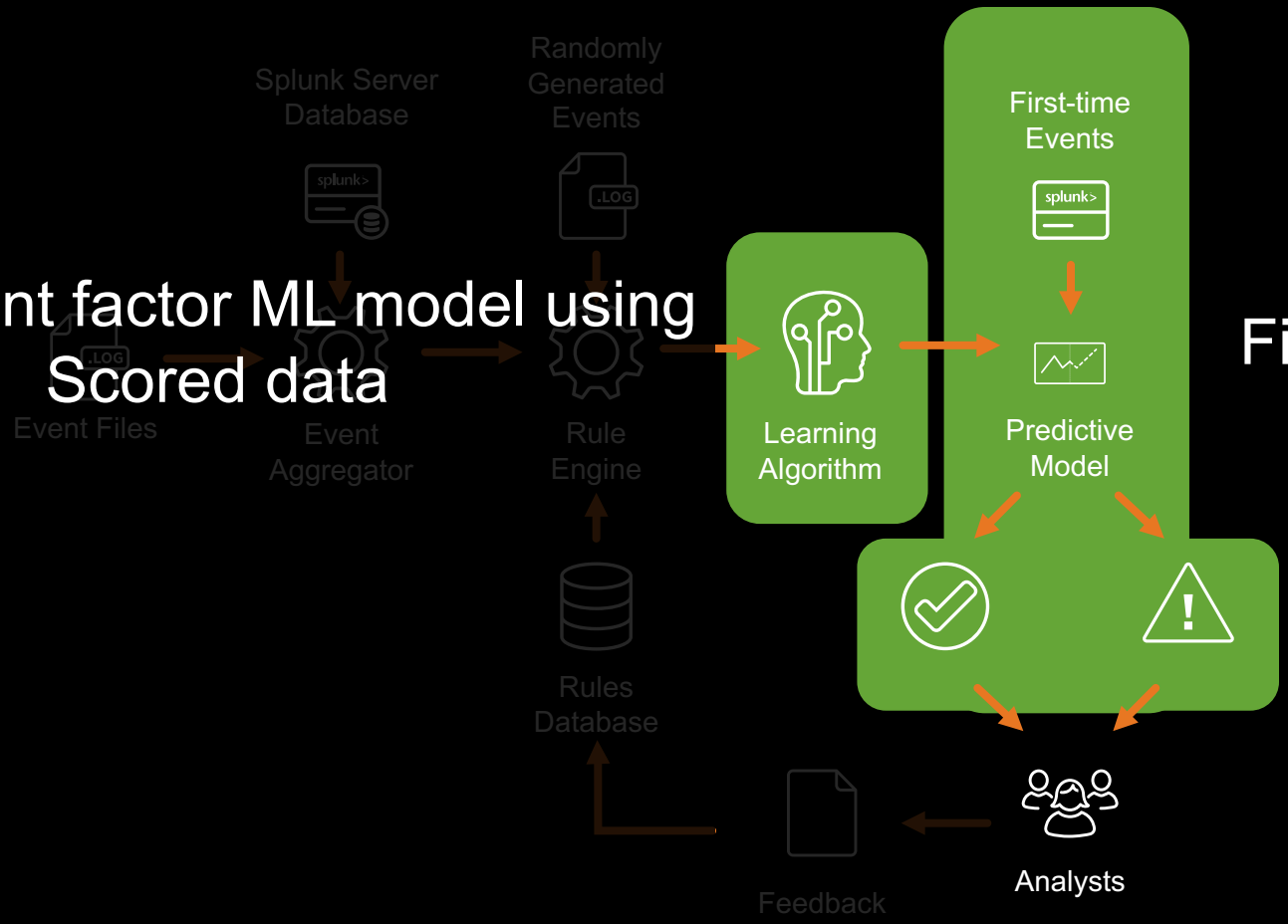
Link ID	User Name	Rule Score	Device Name
1	Bob	100	Buttercup
2	Alice	3	2ndfloorPRT

splunk> .conf18

Machine Learning and Prediction

Step 3 and 4: find model that reconstruct data and use it to predict first time events

Train latent factor ML model using Scored data

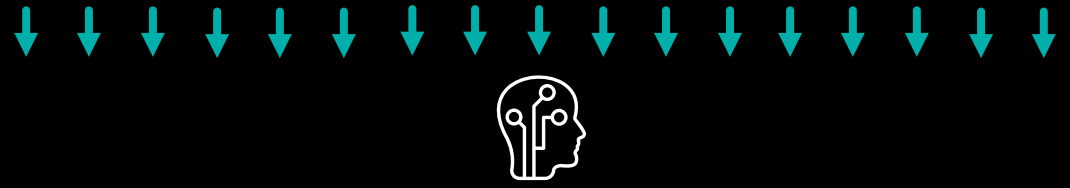


Verify relevance of First-time events using Trained Model

Machine Learning

Step 3: Find a model that can reconstruct the data

Link ID	User Name	Rule Score	Device Name
1	Bob	100	Buttercup
2	Alice	3	2ndfloorPRT



Learning Algorithm

User Name	Latent Factors
Bob	[0.0, 0.5, 0.5]
Alice	[0.8, 0.2, 0.0]

$$f(u, v)$$

Device Name	Latent Factors
Buttercup	[0.0, 0.0, 1.0]
2ndfloorPRT	[0.2, 0.8, 0.0]

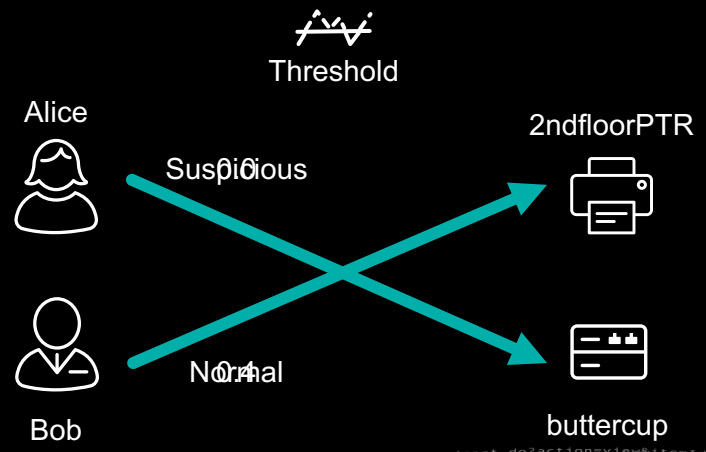
Predictions

Step 4: Predict first time events

User Name	Latent Factors
Bob	[0.0, 0.5, 0.5]
Alice	[0.8, 0.2, 0.0]

Device Name	Latent Factors
Buttercup	[0.0, 0.0, 1.0]
2ndfloorPRT	[0.2, 0.8, 0.0]

$$f(u,m) = u \cdot m = u_0 m_0 + u_1 m_1 + u_2 m_2$$



Results

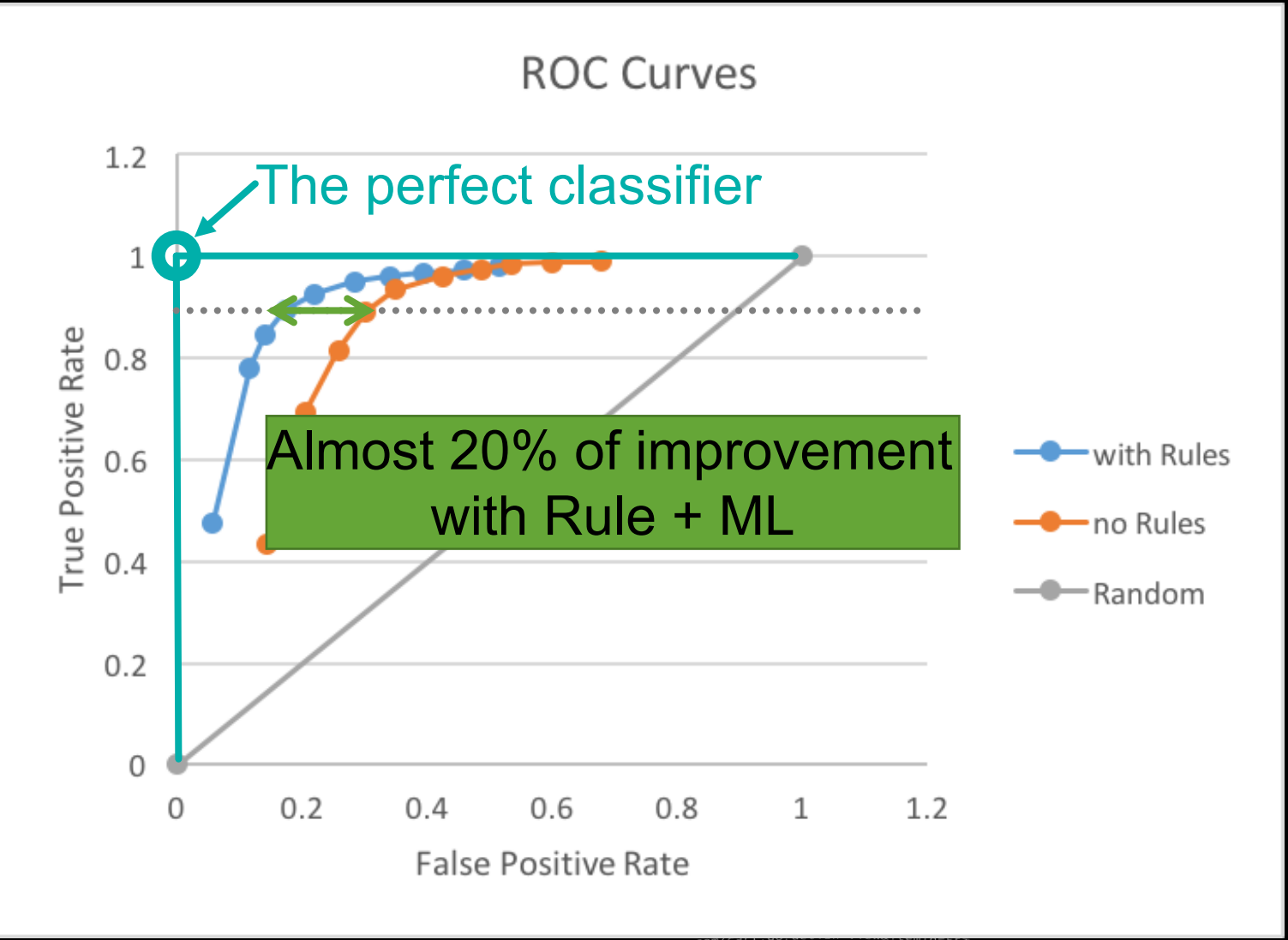
Applying this technique on data

Device Logins

- Login events into devices
  - Users have different roles that define their login behaviors
  - Heterogeneous set of device types
  - Other properties such as employee department, device subdomain are available
  - Distribution of device types, and user roles are biased
- Suspicious accesses happen with low probability
  - Accesses out-of-profile behavior

Pure ML vs Rule + ML

Performance of approach in terms of True and False Positives

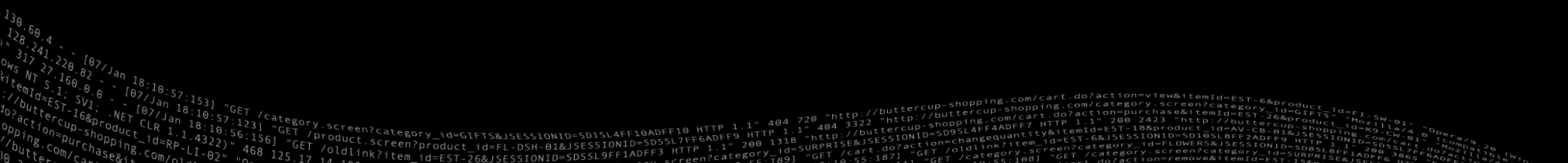


Final Words

Conclusion

Better than rules or machine learning alone

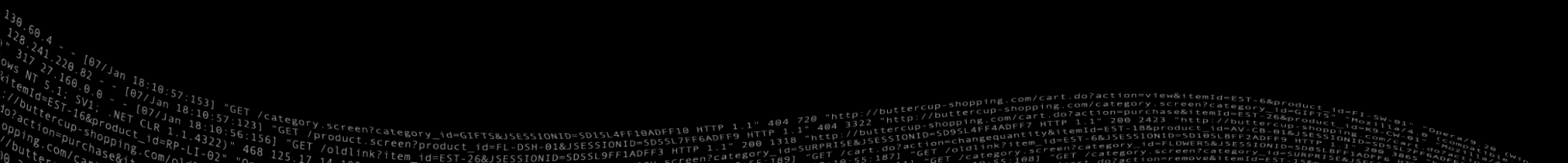
- Reduce of first time events for analyst to look at
- Allows early detection of suspicious activity
- Customizable ML through rules and feedback
- Learns natural behavior not foreseen by rules



Beyond Prediction

Not just predictions of suspicious events

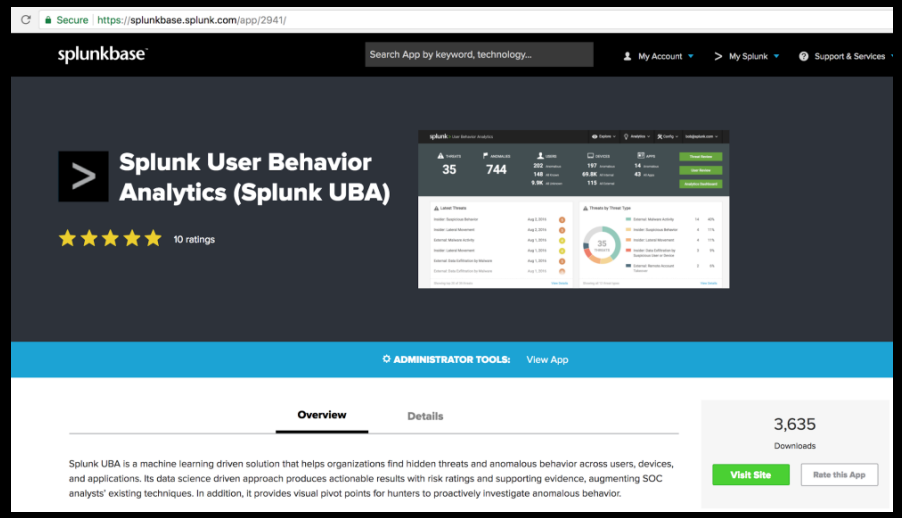
- Latent factors are represented by a vector
  - It encodes behavior and characteristics pretty much like a DNA for entities
- These can be clustered using Splunk MLTK
  - Find behavioral peer-groups of users and devices
  - Define user-device affinity groups
  - And many other interesting insights of your data

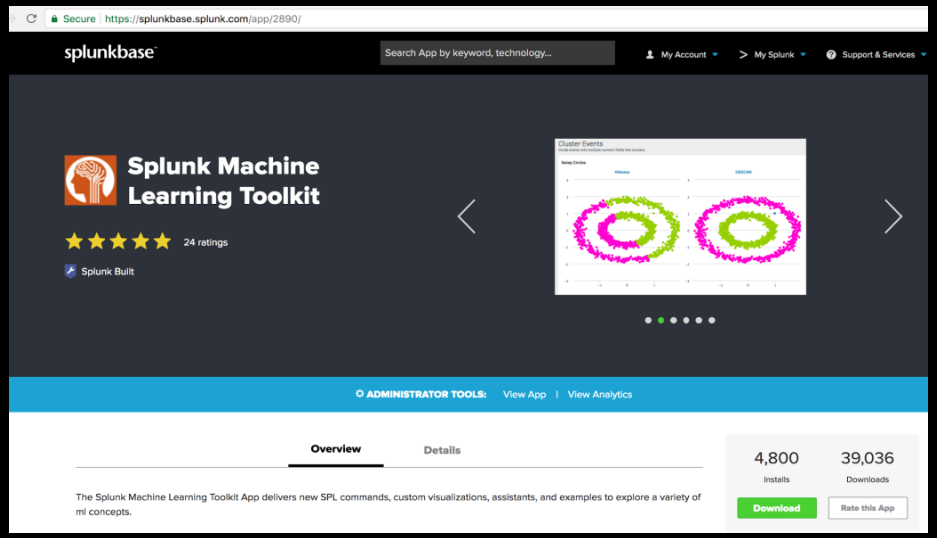


Hands on: Where to go from Here!

Get the power of rules and collaborative filtering algorithms

Splunk User Behavior Analytics
Splunk-UBA





Splunk Machine Learning Toolkit
Splunk-MLTK

splunk> .conf18

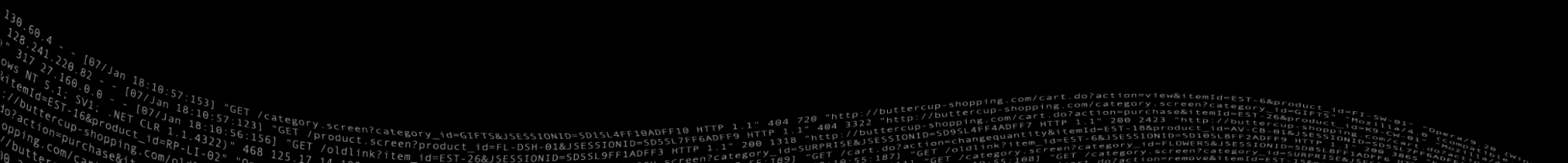
Q&A

Ignacio Bermudez Corrales| icorrales@splunk.com

First Time Events in Real Life

Some are normal, some can be risky

- Machine Accesses
  - IT workers frequently update machines. One updates a QA testing VM, is this risky?
- Cloud Storage File-sharing
  - Files are download or previewed once, some may be read by the “wrong” people
- Private Code Repositories
  - An employee’s account got compromised and cloned projects with sensitive intellectual property
- Badge Accesses
  - Some worker entered the servers room of his office out of working hours



We Want Early Detection

The risk of first time event and why you should care

- Event linked to suspicious activity can happen only once
  - Attacker spear downloads single confidential file to local hard drive
  - Suspicious activity happens once and there are not after events to make a decision
    - Volume of data download, number of devices accessed, number of repos cloned, etc.
- Detecting suspicious activity based on post-events can be risky
  - Collecting post-events can take time, can we make an early decision?
- Many normal operations happen for a first time
  - Employee access some colleague's dev machine, architect downloads confidential document regarding product design

First Time Events in Real Life

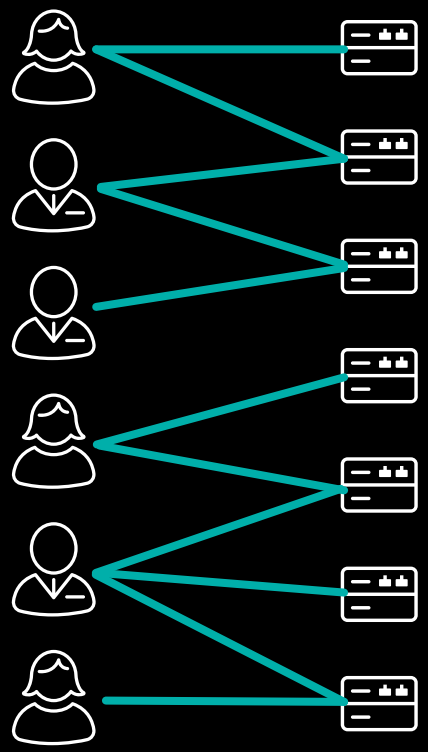
Some are normal, some can be risky

- Users access similar set of machines on a daily basis
  - IT workers frequently update machines. One updates a QA testing VM, is this risky?
- Employees share files in some cloud storage system
  - Files are download or previewed once, some may be read by the “wrong” people
- A git repository where all developers have read access
  - An employee’s account got compromised and cloned projects with sensitive intellectual property
- Company keeps record of badge accesses
  - Some worker entered the servers room of his office out of working hours

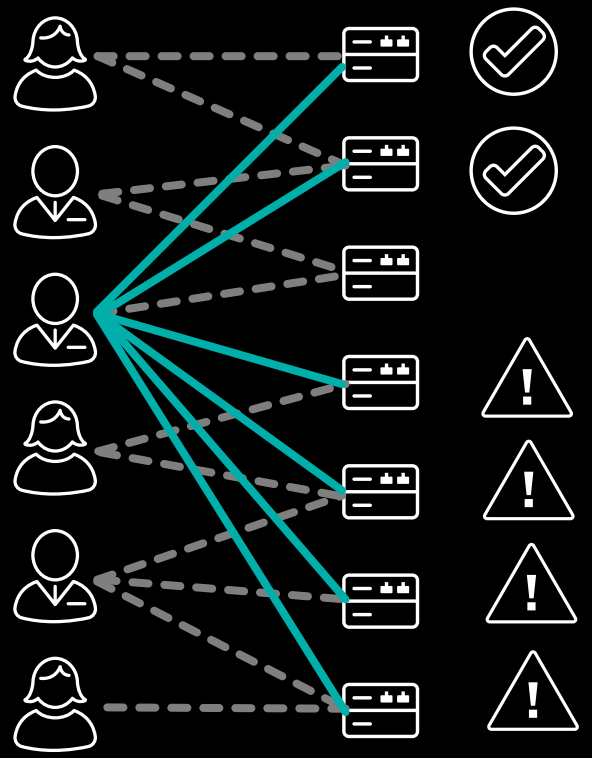
Machine Accesses

Do all first time accesses look suspicious?

Historical Accesses



First Time Accesses



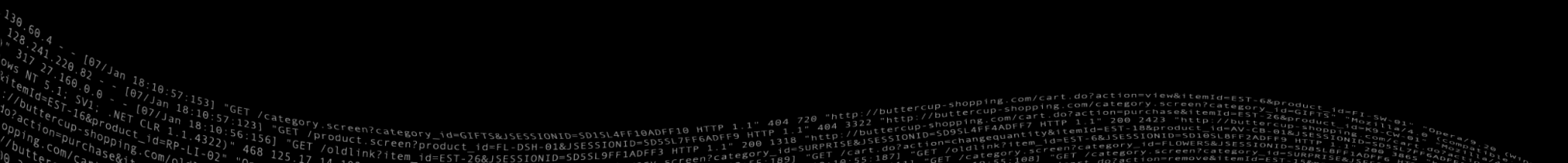
Normal Operations?

Suspicious Accesses?

Examples of First Time Events

- First time …
  - A user access a machine
  - A user enters a room in a building
  - A user opens a file in a private cloud
  - A user pulls code from a git repository
  - A user logins from a particular location
  - An employee mails another employee
  - And many other examples

- First time events can be pretty common



First Time Event Handling

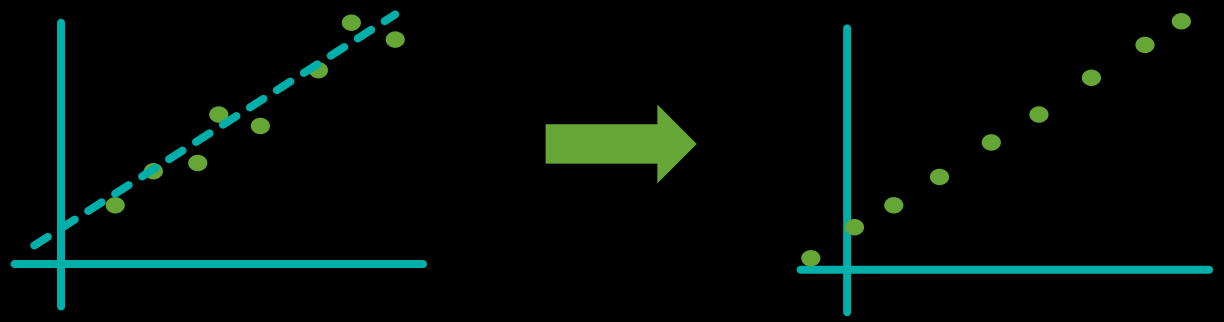
How to deal with first time events?

- Conservative Approach
  - All first time events are suspicious and worth looking at one by one
  - In scenarios where the number of first time events is huge this requires excessive resources
- Laissez-faire Approach
  - We do not care about first time events
  - Easy to implement, but won't catch any concerning first time event
- Use Rules
  - They can cover low hanging fruits, but can't cover corner cases
- Use Machine Learning
  - Learns from historical patterns in the data, but can't learn from what haven't happened
- The answer resides between the last two!

Some Machine Learning Background

Things we should keep in mind through the presentation

- Machine learning algorithms approximate numerical data with a function
  - History of events have to be represented as numbers to learn
  - Linear regression: Given data points (x, y) find a and b such that y' = a x + b can **reconstruct** the evidence



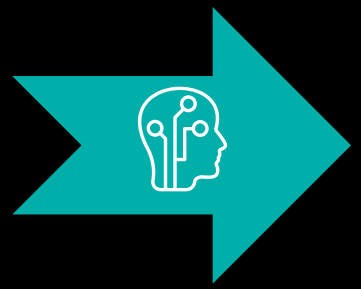
We need to model our problem **numerically** and we will find **parameters** that can reconstruct the evidence

Machine Learning Generalizes

ML allows reconstruction of the data

Historical Data - Evidence

	1	1					
		1	1				
			1				
				1	1		
					1	1	1
							1



ML Reconstruction of Data

	1	1	0.8	0.1	0.1	0.1	0.1
	0.8	1	1	0.1	0.1	0.1	0.1
	0.8	0.8	1	0.1	0.1	0.1	0.1
	0.1	0.1	0.1	1	1	0.8	0.8
	0.5	0.5	0.5	0.8	1	1	1
	0.1	0.1	0.1	0.5	0.8	0.8	1

The objective of the ML algorithm is to reconstruct the evidence
With the minimum error possible using an evaluation function

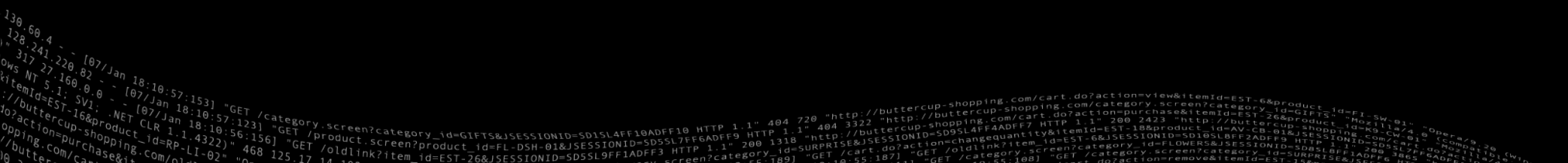
Taming ML with Rules - Review

Passing down knowledge to machine learning algorithm

- Rules:
  - Embed human knowledge
  - Map event properties into numerical signal
  - Consume external properties of event entities
  - May have low coverage
  - Complements what can be seen in the history
  - Don’t generalize
- Passing them down
  - Generate random events
  - If random event hit a rule add them to history
  - Control the number of random events to avoid overfitting

Coverage of Rules

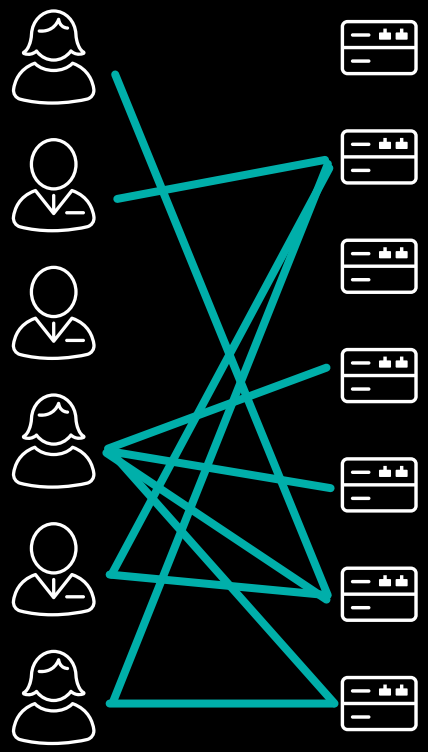
						1	
		1					
				1	1	0	0
		0				1	
		0				1	



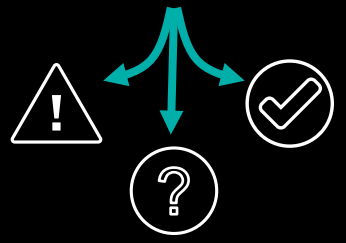
splunk> .conf18

Taming ML with Rules - Review

Passing down knowledge to machine learning algorithm



if(event == condition)



Coverage of Rules

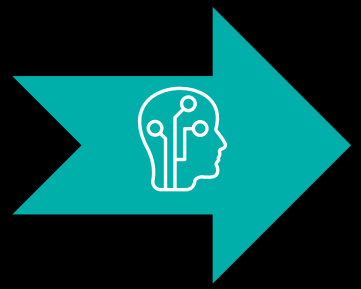
						1	
		1					
				1	1	0	0
		0				1	
		0				1	

Machine Learning Generalizes

ML allows reconstruction of the data

Historical Data - Evidence

	1	1					
		1	1				
			1				
				1	1		
					1	1	1
							1



ML Reconstruction of Data

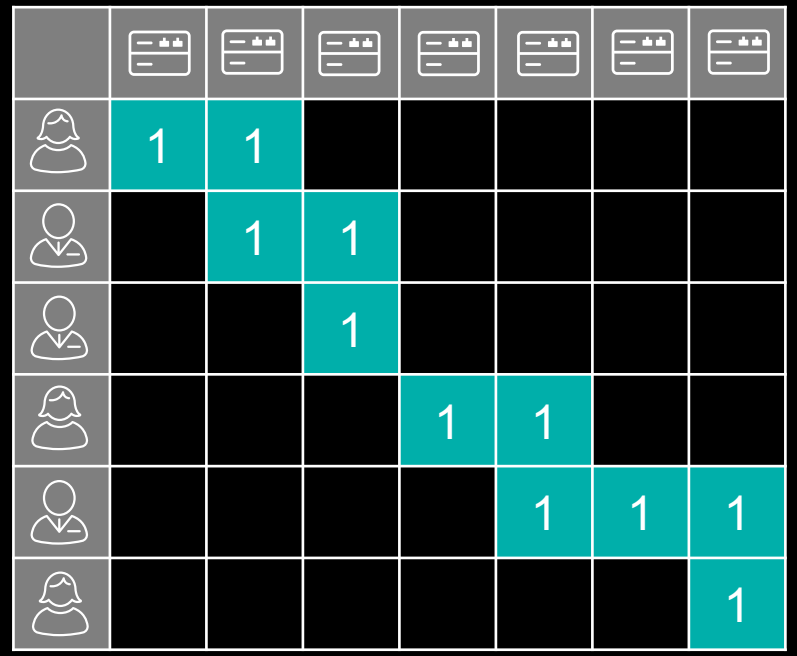
	1	1	0.8	0.1	0.1	0.1	0.1
	0.8	1	1	0.1	0.1	0.1	0.1
	0.8	0.8	1	0.1	0.1	0.1	0.1
	0.1	0.1	0.1	1	1	0.8	0.8
	0.5	0.5	0.5	0.8	1	1	1
	0.1	0.1	0.1	0.5	0.8	0.8	1

This is analogous to the linear regression problem when $y' = a x + b$
Reconstructs the Data

Machine Learning Generalizes

How a machine learning model predicts first time events

Historical Data - Evidence



ML Reconstruction of Data

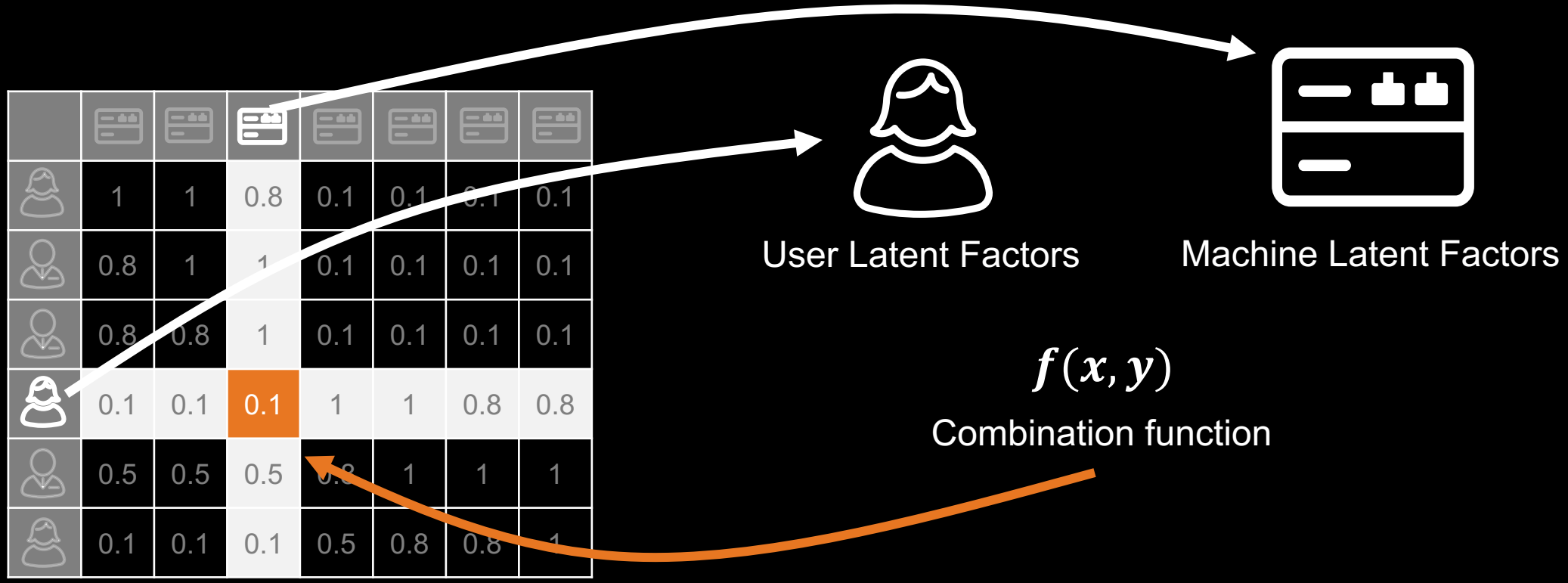
	1	1	0.8	0.1	0.1	0.1	0.1
	0.8	1	1	0.1	0.1	0.1	0.1
	0.8	0.8	1	0.1	0.1	0.1	0.1
	0.1	0.1	0.1	1	1	0.8	0.8
	0.5	0.5	0.5	0.8	1	1	1
	0.1	0.1	0.1	0.5	0.8	0.8	1

This is analogous to the linear regression problem when $y' = a x + b$

Reconstructs the Data

Machine Learning Generalizes

How a machine learning model predicts first time events



Analogous to the linear regression problem **y' = a x + b**

Latent factors are the parameters (a, b) and the function combines them

How Data is Reconstructed

Latent factor models

Machine Latent Factors (y)

User Latent Factors (x)

	1	1	0.8	0.1	0.1	0.1	0.1
	0.8	1	1	0.1	0.1	0.1	0.1
	0.8	0.8	1	0.1	0.1	0.1	0.1
	0.1	0.1	0.1	1	1	0.8	0.8
	0.5	0.5	0.5	0.8	1	1	1
	0.1	0.1	0.1	0.5	0.8	0.8	1

$$f(x, y) = 0.1$$

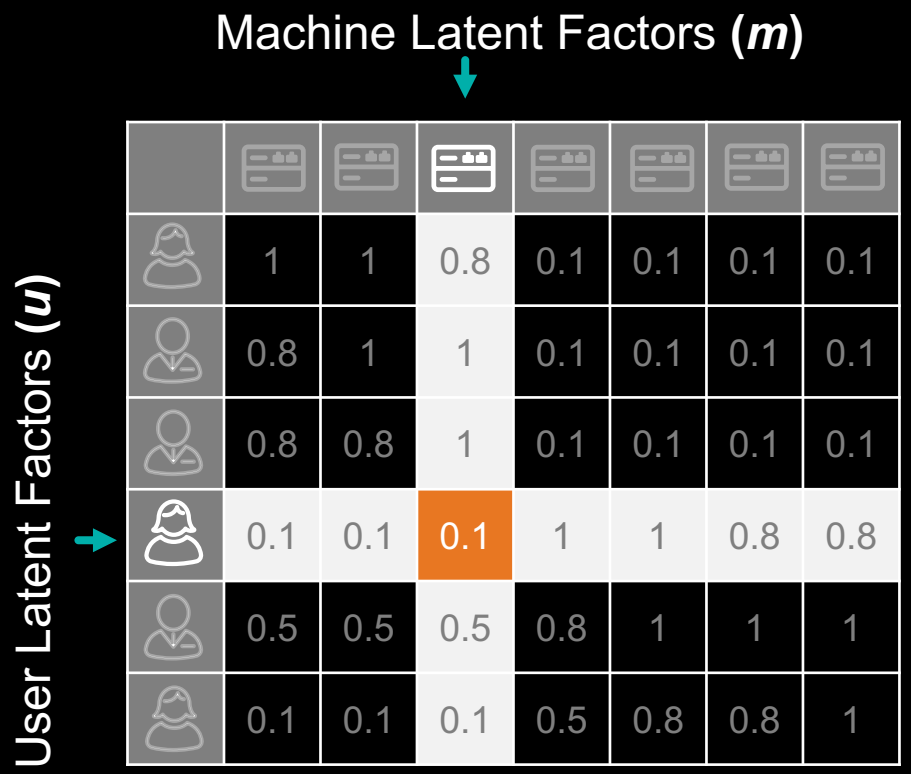
Evaluation function

In the machine learning training phase
we find **latent factors** parameters,
that combined with a **function**,
we can recreate the values of the matrix
(remember, similar to y' = a x + b)

splunk> .conf18

How Data is Reconstructed

Latent factor models



Latent factors **(*u, m*)** are real valued vectors that minimize the reconstruction error made by the evaluation function

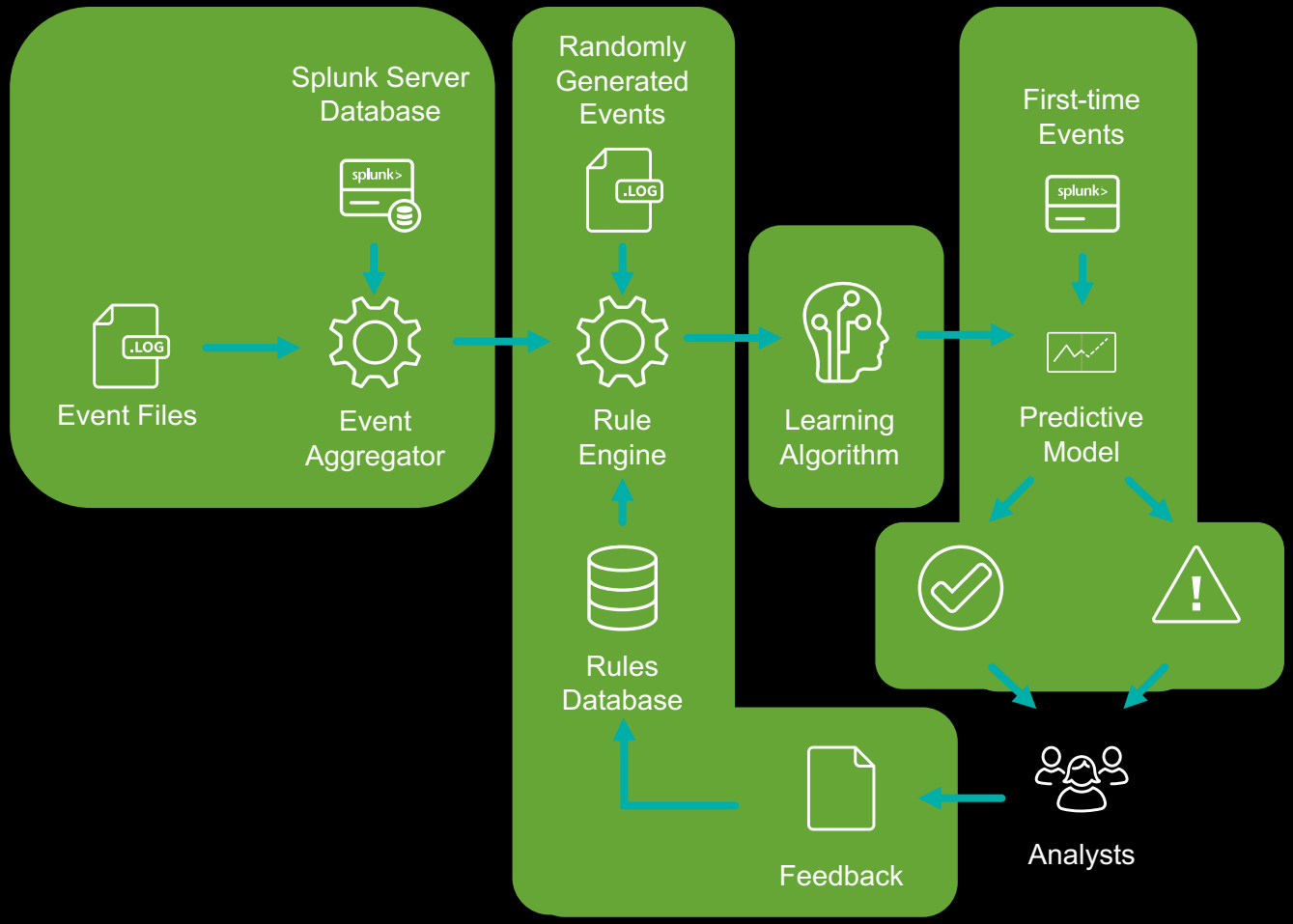
$$f(u, m) = 0.1$$

Evaluation function

This is analogous to linear regression where we find parameters (*a, b*) that minimize the error of y' = a x + b

Architecture

4 steps



Event Processing

Step 1

Input

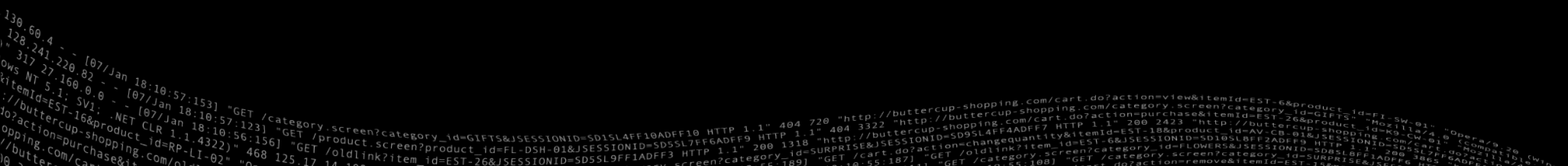
- Event Logs
- External information about entity properties

Goal

Convert single event logs into an annotated graph with event properties

Output

Table of aggregated events



splunk> .conf18

Event Processing

Step 1

- Goal
  - Convert single event logs into an annotated graph with event properties
- Input
  - Event logs
  - External information about the entities
- Output
  - Table where each row is an aggregation of events between two entities
  - For each row there are properties attached to them

Link ID	User Properties	User Name	Events Summary	Device Name	Device Properties
1	Peergroup: 1 OU: Engineering	Bob	SSH duration 2h	Buttercup	Type: VM Domain: dev
2	Peergroup: 10 OU: Finance	Alice	IPP	2ndfloorPRT	Type: Printer Domain: iot

Rule Engine

- Goal
  - Convert links into numerical signals (scores)
- Input
  - Table of aggregated events
  - Randomly generated event aggregations
  - Rules database
  - Previous analyst feedback
- Output
  - Table with scored links

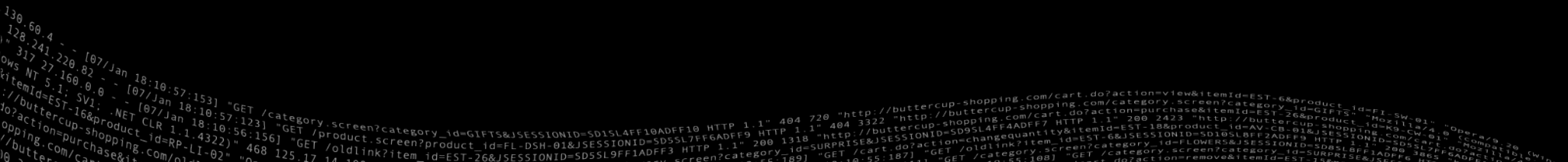
Link ID	User Name	Rule Score	Device Name
1	Bob	100	Buttercup
2	Alice	3	2ndfloorPRT

Machine Learning Algorithm

- Goal
  - Generate latent factors for entities that allow reconstruction of the data
- Input
  - Real valued links with labeled entity identifiers
- Output
  - Evaluation function that can operate on latent factors (given by the algorithm)
  - Latent features for each entity

User Name	Latent Factors
Bob	[0.0, 0.5, 0.5]
Alice	[0.8, 0.2, 0.0]

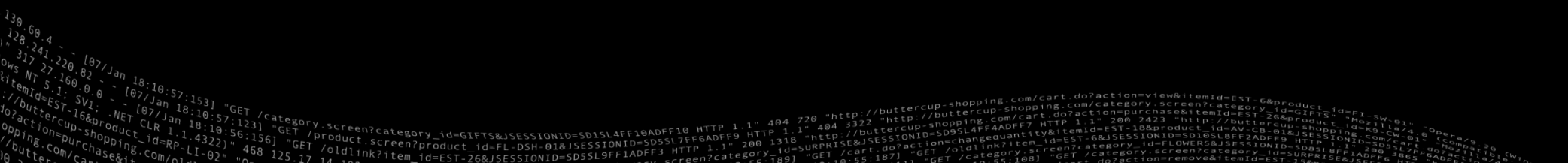
Device Name	Latent Factors
Buttercup	[0.0, 0.0, 1.0]
2ndfloorPRT	[0.2, 0.8, 0.0]



Predictions

Step 4

- Goal
  - Predict anomalous first time events
- Input
  - Database of entity latent factors
  - Function that operates on latent factors
  - Anomaly threshold
- Output
  - Labeled anomalous first time events



Example

User factors lookup table

User Name	Latent Factors
Bob	[0.0, 0.5, 0.5]
Alice	[0.8, 0.2, 0.0]

Device factors lookup table

Device Name	Latent Factors
Buttercup	[0.0, 0.0, 1.0]
2ndfloorPRT	[0.2, 0.8, 0.0]

Evaluation function

$$f(x, y) = x \cdot y = x_0 y_0 + x_1 y_1 + x_2 y_2$$

Predicted scores

	Buttercup	2ndfloorPRT
Bob	0.5	0.4
Alice	0.0	0.8

Threshold

Predictions

	Buttercup	2ndfloorPRT
Bob	Normal	Normal
Alice	Suspicious	Normal

Predictions

Step 4: Predict first time events

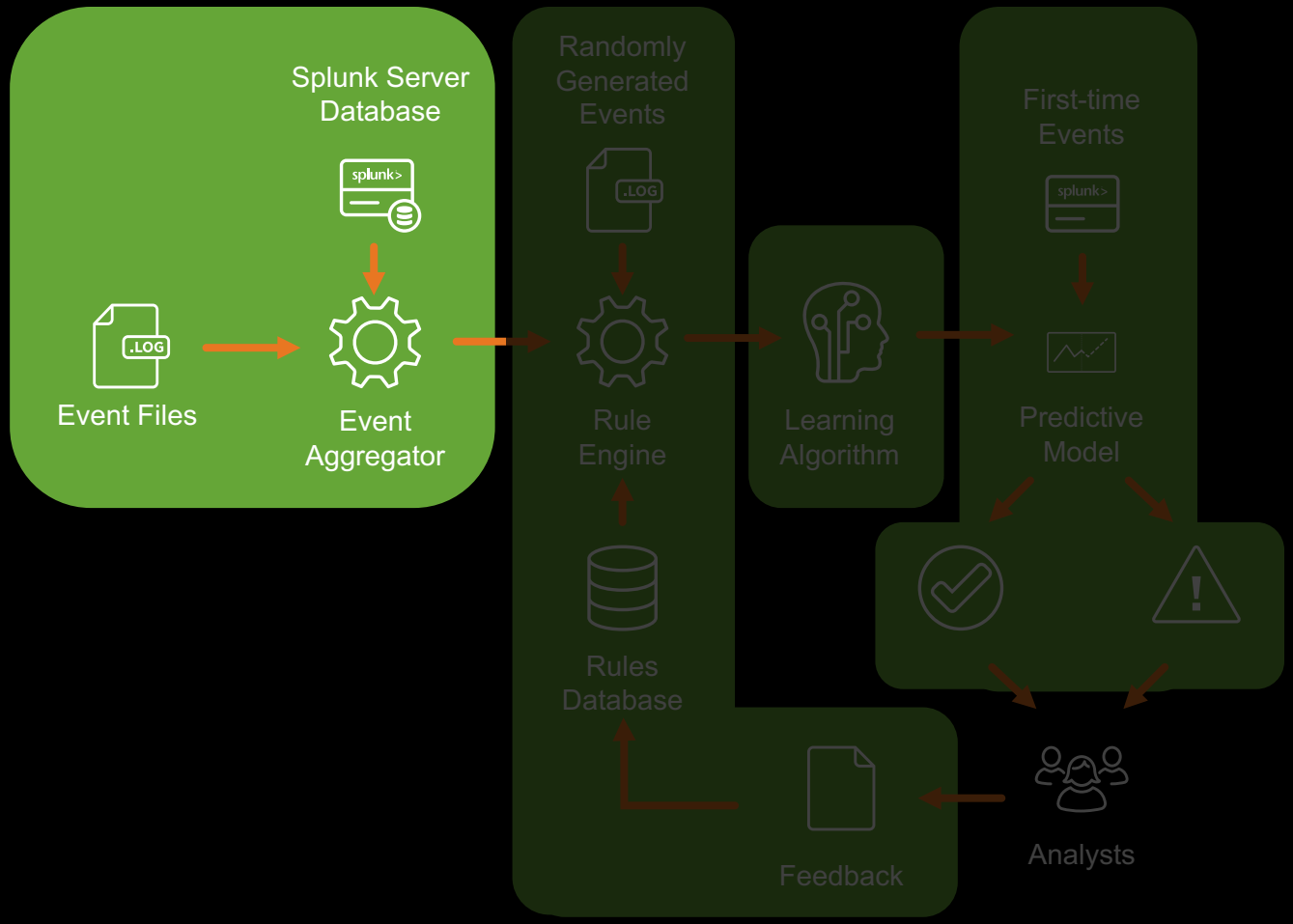
User Name	Latent Factors
Bob	[0.0, 0.5, 0.5]
Alice	[0.8, 0.2, 0.0]

Device Name	Latent Factors
Buttercup	[0.0, 0.0, 1.0]
2ndfloorPRT	[0.2, 0.8, 0.0]

$$f(x,y) = x \cdot y = x_0y_0 + x_1y_1 + x_2y_2$$

Event Processing

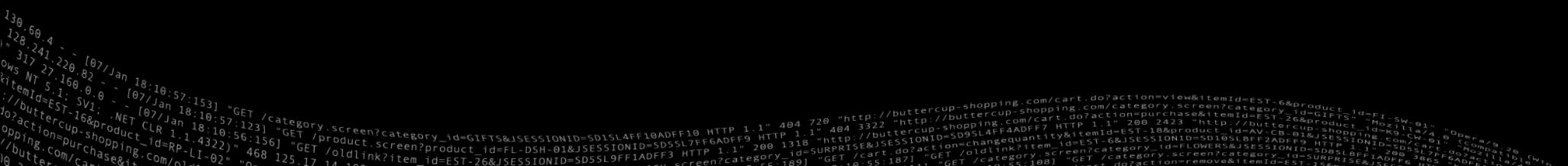
Step 1: Build a summary of information for the rule engine



Event Processing

Step 1

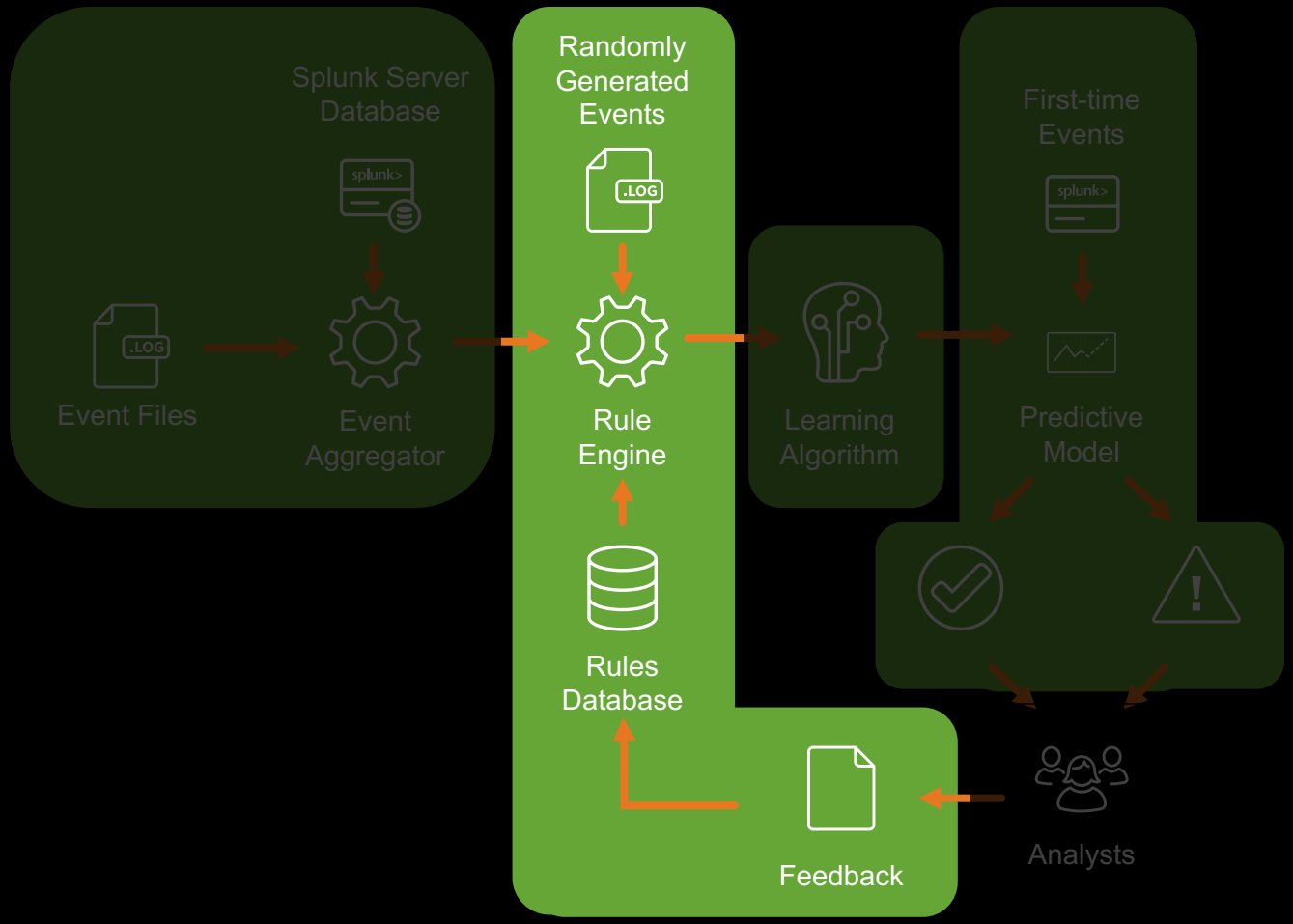
Input	Goal	Output
• Event Logs • External information about entity properties	Provide as much information as possible to the rules to make decisions	Table of aggregated events



splunk> .conf18

Rule Engine

Step 2: Convert event summaries into numerical signals



Rule Engine

Step 2

Input

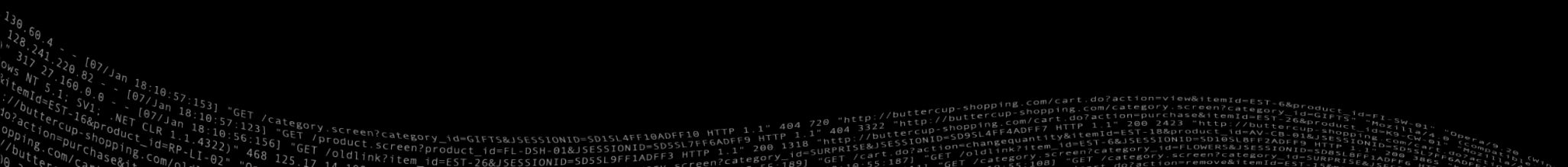
- Table of aggregated events
- Randomly generated events
- Rules database
- Previous feedback

Goal

Convert aggregations into numerical signals, so they can be learnt by a Machine Learning algorithm

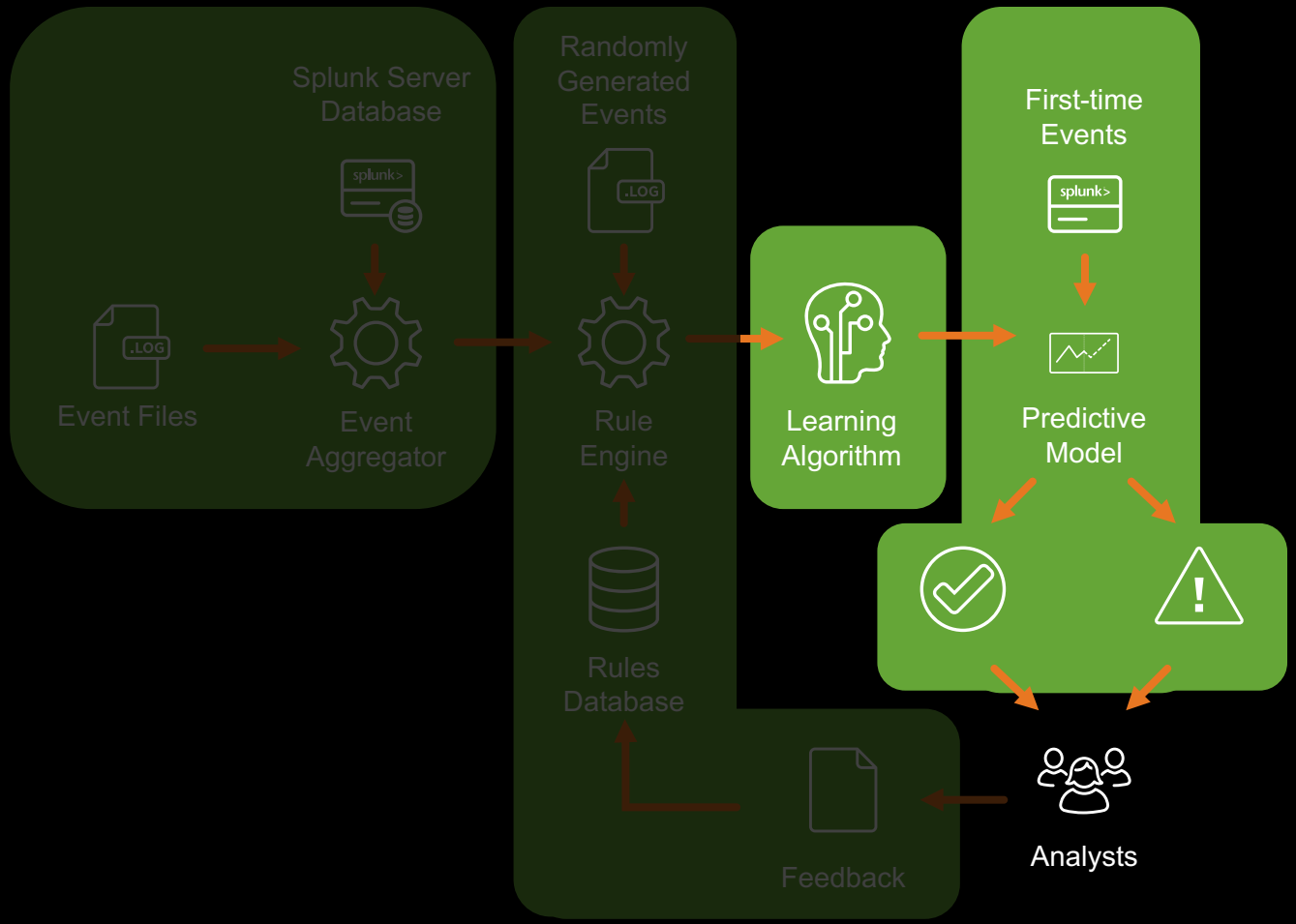
Output

Table of scored links



Machine Learning and Prediction

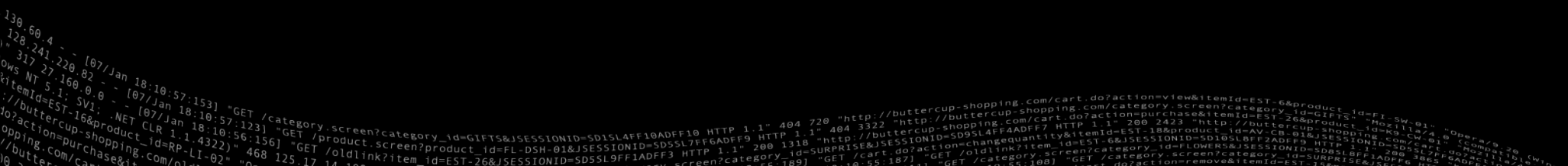
Step 3 and 4: find model that reconstruct data and use it to predict first time events



Machine Learning

Step 3

Input	Goal	Output
Table of scored links	Find a model that can reconstruct the data	• Evaluation function • Latent factors for each entity

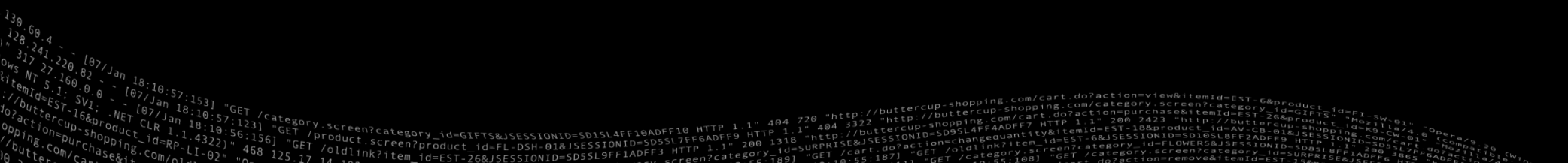


splunk> .conf18

Predictions

Step 4

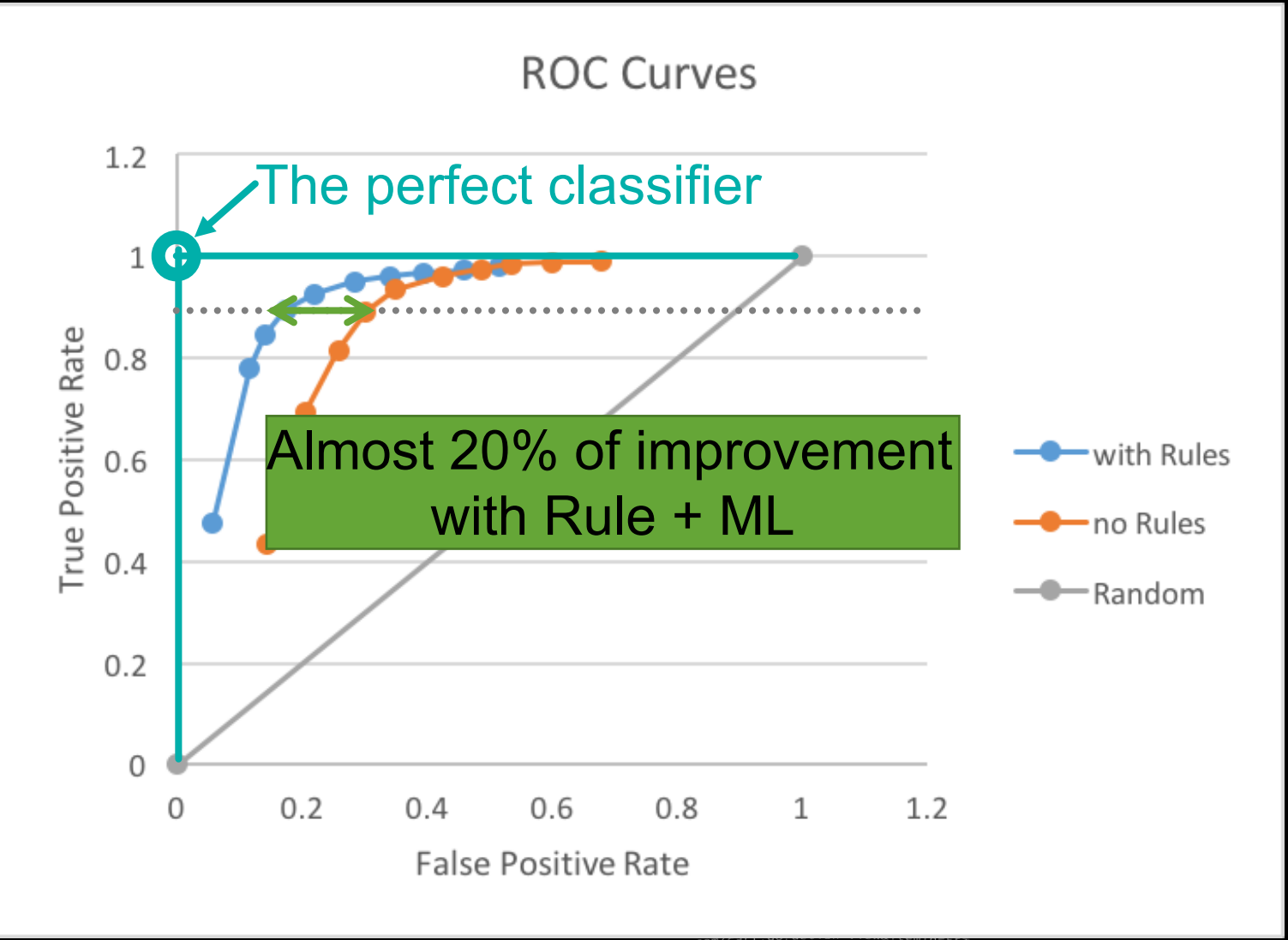
Input	Goal	Output
• Table of aggregated events • Randomly generated events • Rules database • Previous feedback	Predict anomalous first time events	Labeled first time events



splunk> .conf18

Pure ML vs Rule + ML

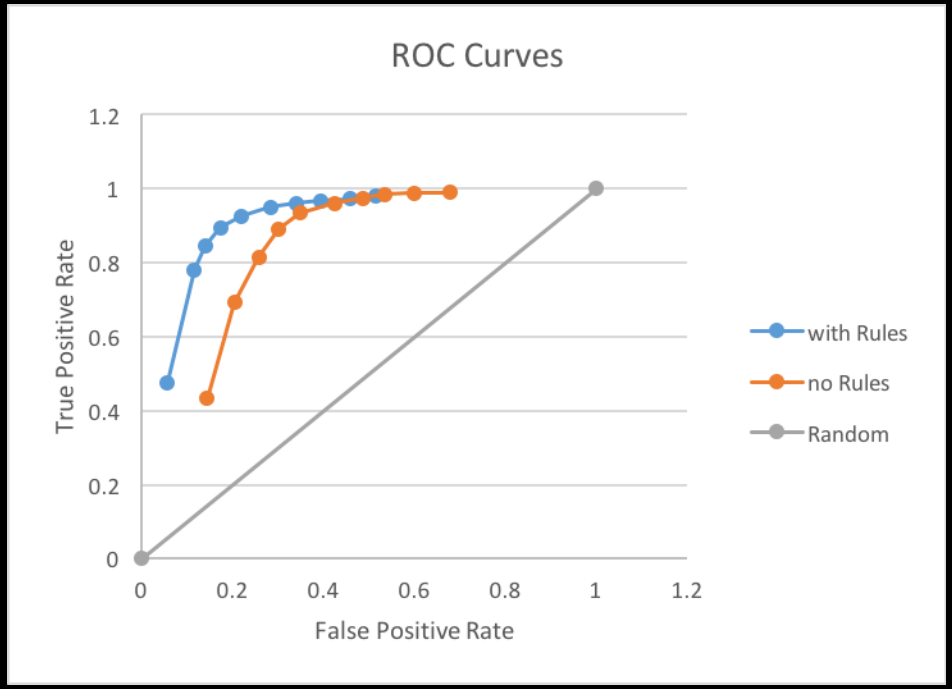
Performance of approach in terms of True and False Positives



Pure ML vs Rule + ML

Performance of approach in terms of True and False Positives

- Compare approach using and not using rules
- Rate of TP/FP by moving the detection threshold
- Rules improve overall performance
- Same TP rate lesser FP
  - TP = 0.9, FP = 0.2 with rules
  - TP = 0.9, FP = 0.35 without rules



Conclusion (make 2 slides)

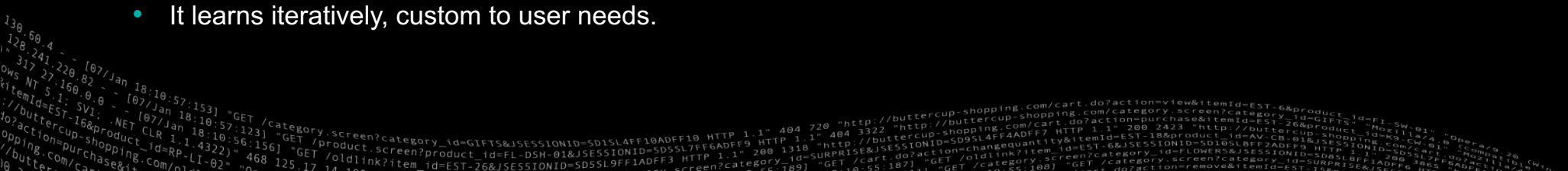
How this is implemented on Splunk. Where you can find it in splunk. A second slide is how this is gonna help you, compare to other products if possible (competitors).

- This can be implemented using Alternating Least Squares (ALS) on Spark
  - Can also be implemented in Python with some specific packages

- UBA suspicious first Box accesses use this approach
  - Captures when a user access a resource for the first time
  - Uses Active Directory extracted peer-groups
  - Rules based on file properties

- Not just prediction of suspicious first time event
  - Latent factors as “DNA” for entity behavior or characteristic
  - Latent factors … explain .. Like a DNA
  - Enables further behavioral clustering of entities

- What this all means for a business (the impact, why this is important)
  - Save analyst time
  - It learns iteratively, custom to user needs.

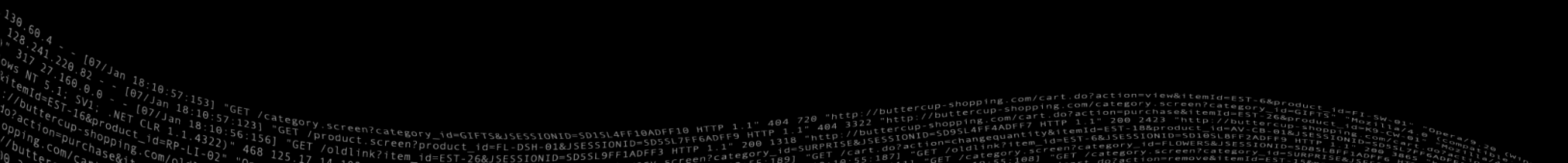


splunk> .conf18

Conclusion

Better than rules or machine learning alone

- Allows early detection of suspicious activity
  - Reduces the volume of first time events the analyst has to look at
- Customizable through rules and feedback
  - Analysts can impact ML model by providing feedback on specific events or by writing rules
- Learns natural behavior not foreseen by rules
  - Collaborative filtering extrapolates from historical interactions



Hands on

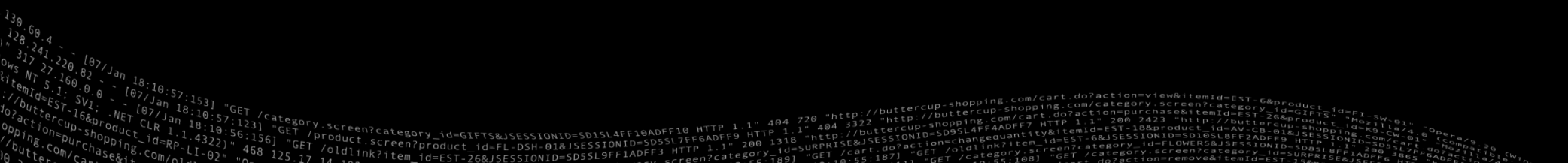
Get the power of rules and collaborative filtering algorithms

- Splunk UBA
  - Currently used for detecting suspicious access to resources in Box
  - Leverages Spark ML collaborative filtering API
- Splunk Machine Learning Toolkit (MLTK)
  - Adding more algorithms from Spark ML soon
- Not just predictions of suspicious events
  - Latent factors represent a vector encoding behavior, pretty much like a DNA for entities
  - They can be used to cluster entities based in their behavior to find interesting insights

Hands on

Get the power of rules and collaborative filtering algorithms

- Splunk UBA
  - Currently used for detecting suspicious access to resources in Box
  - Leverages Spark ML collaborative filtering API
- Splunk Machine Learning Toolkit (MLTK)
  - Adding more algorithms from Spark ML soon



splunk> .conf18