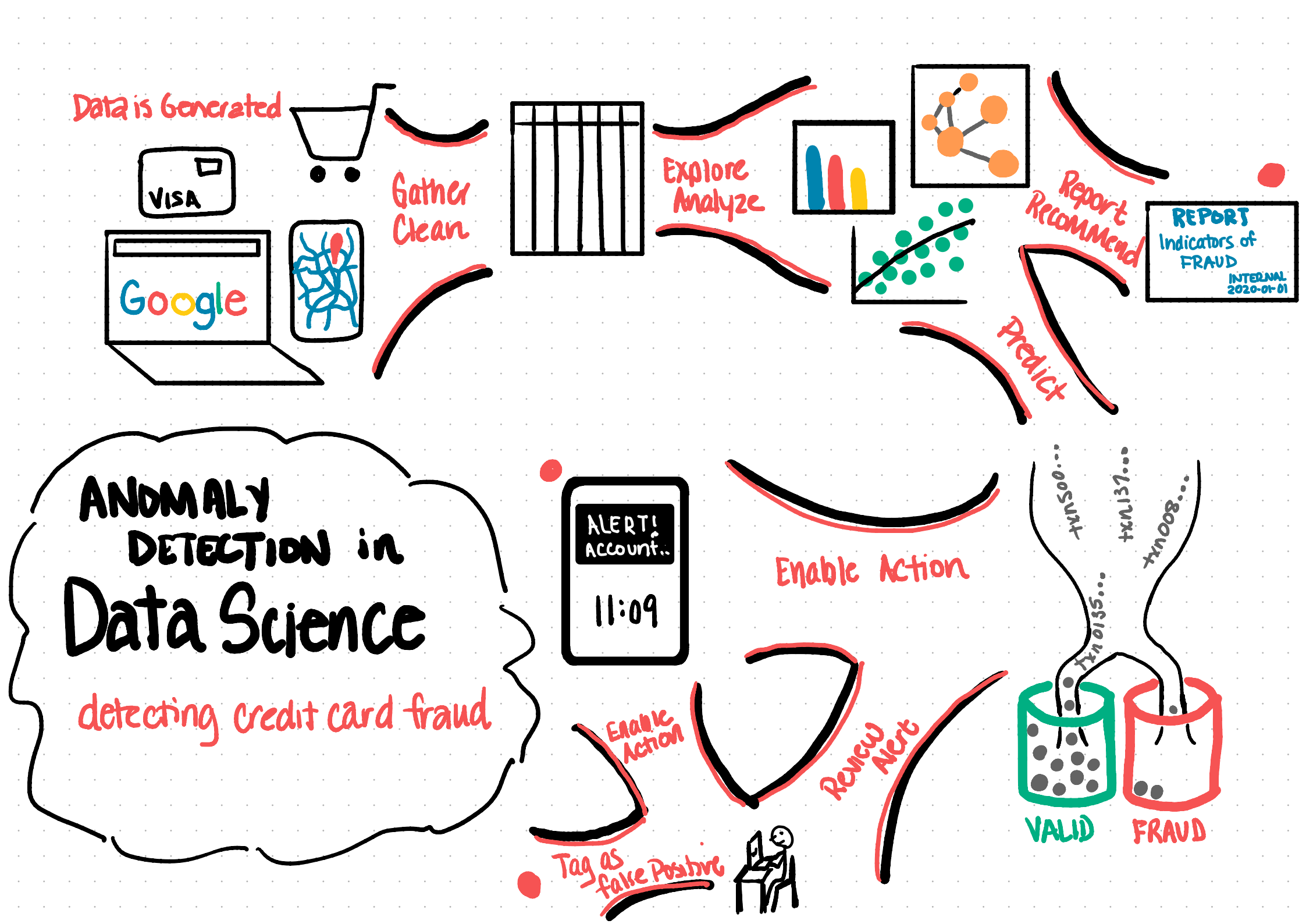
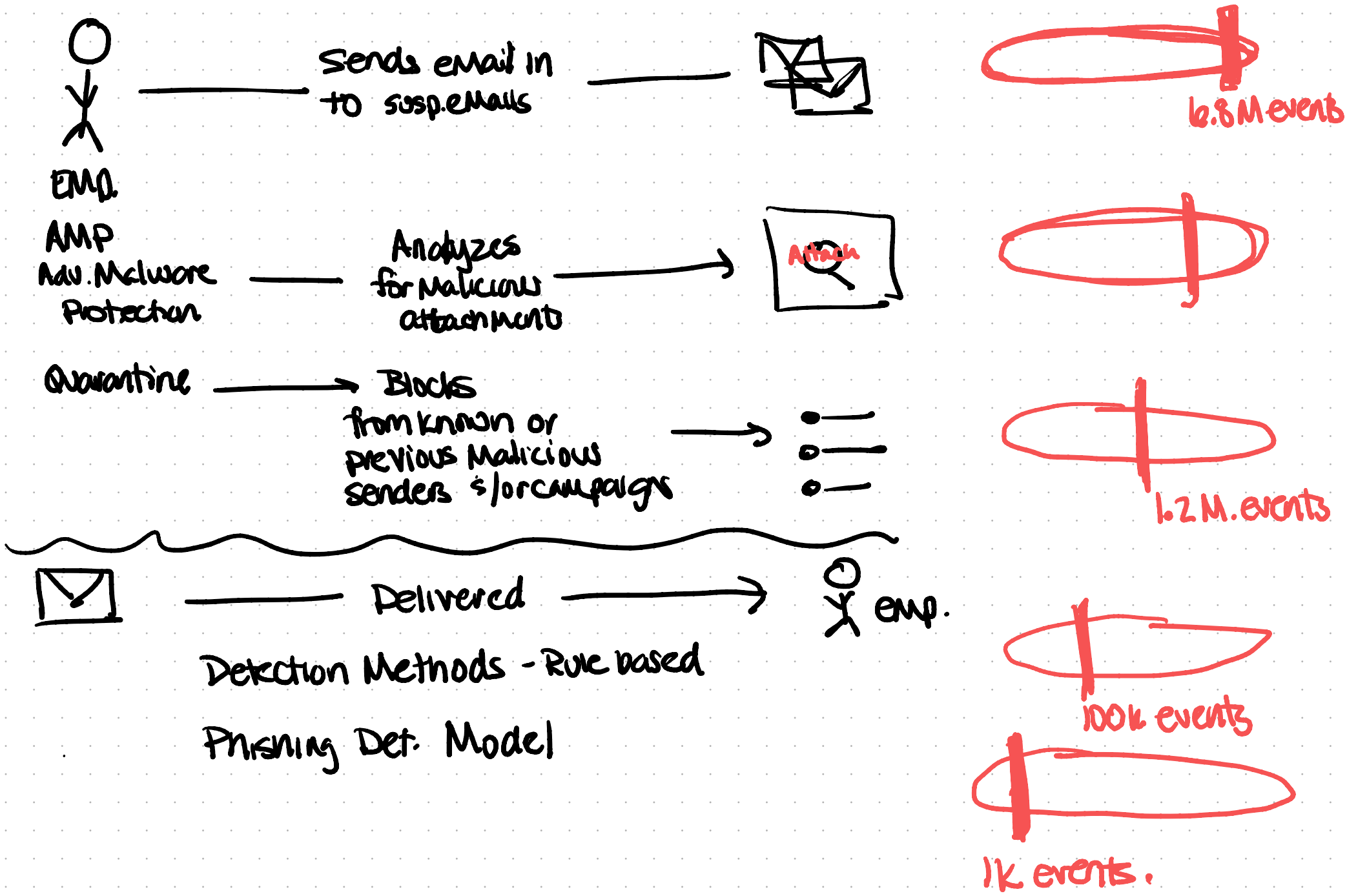
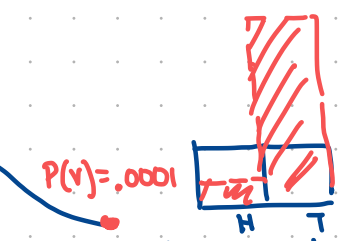

Data is Generated
VISA
Google
Gather
Clean
Explore
Analyze
Report
Recommend
REPORT
Indicators of
FRAUD
INTERNAL
2020-01-01
Predict
ANOMALY
DETECTION in
Data Science
detecting credit card fraud
ALERT!
Account..
11:09
Enable Action
Review Alert
Enable
Action
Tag as
false Positive
txn500...
txn137...
txn800...
txn0135...
VALID
FRAUD


Sends email in to susp.emails
EMP.
AMP
Adv. Malware Protection
Analyzes for Malicious attachments
Attach
Quarantine
Blocks from known or previous Malicious senders s/or campaigns
Delivered
EMP.
Detection Methods - Rule based
Phishing Det. Model
6.8 M events
1.2 M. events
100k events
1K events.

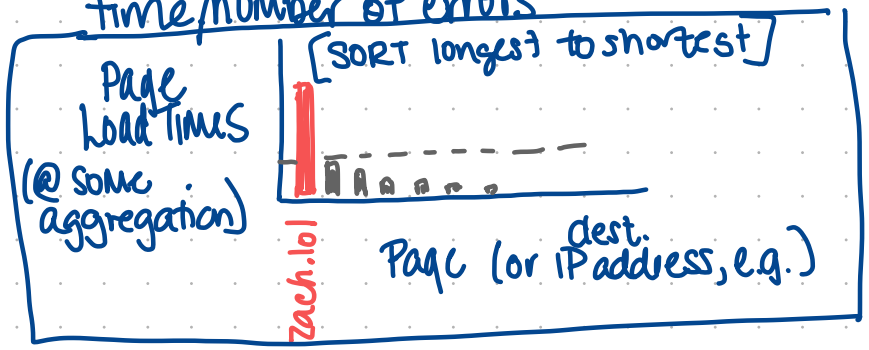
Anomaly Detection using Probabilities



How it works: **Visualization**

Stats + Rules: use probability distributions compute prob./expected values & use Rules to Rank/score/highlight those most likely to be anomaly.

How it's used: **dashboard**: sysadmins monitoring system performance like uptime/downtime/response time/number of errors



Alert: email alert - "We've noticed significant increase in response time for zach.lol. You might want to activate some load balancers if an increase in traffic is expected and wanted traffic."

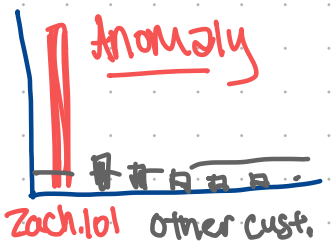
flag on a report "abnormal", such as a "blood test."

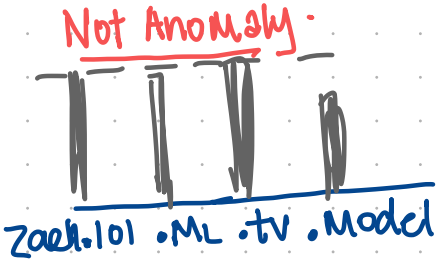
Test	Range	Result	Note
TSH	(.4-4.0)	11.6	Abn.(H)

Pros: [can be]
Simple to implement & for end users to understand what's going on, or why something was flagged.
Can also be very complicated, but still not very computationally expensive

Cons: When the data contains a large amount of noise or variance, this can be challenging without adding extra Rules & layers for filtering.
More dimensions ⇒ more challenges.

Example of challenge





★ Depends on your use case, needs

Time Series Anomaly Detection

CASE Study: Detect data exfiltration.

Planning: How is data most likely to be exfiltrated (low hanging fruit first!)

INSIDE ACTOR	vs.	INTRUDERS
quickly or slowly over time		
to one location		to many locations
to external personal email, cloud drive apps		through ftp, AWS, remote servers.
using common clients		using "off" clients
from 1 user		from 1 or many users.
...		...

What data will be useful to capture some of the above?

- Amount of data going out (GB)
- Frequency of data going out (# of events)
- When data is going out (time of day, day of week)
- Source (IP / URL / email / region (from IP)
- Destination (IP / URL / email / region (from IP)
- "Door" it's going out (Client)
- User (user id, device MAC address, email address

What might we see?

- Source IP location NOT where user is located
- Destination IP never seen before Reg

Time Series Anomaly Detection

CASE Study: Detect data exfiltration.

Planning:
- Get in the Mind of the threat actor (intruder or insider)
- What data can help show us these actions?
- → What anomalies might we see?
- What challenges might we run into? filtering out noise, trying to address all options at once, amount of data-ability to compute
- What decisions will we need to make? When to investigate, prioritize among other alerts w/ e to focus 1st ...

email, torrent, cloud drives, print, photos, remote server (hidden creative domain).

Attributes [dest. IP, URL, Domain, email
source IP, User, email

Time [daily, Night/day, day of wk, Weekday, weekend]

User ID – dest. (size)

Anomaly

↑ Frequency to an IP by a user

Inc. outgoing emails w/ attachments
↑ to personal email / freemail domain

Data going to Rare TLD .ru, .xyz

Measure
- Bandwidth
- Frequency # of POST >_
- # of emails w/ Attachments
- types of files.

Time Series Anomaly Detection

Semisupervised ML

*Measure

- GB out
- #of POST events > X MB.
- # emails with attachments

+Time Period

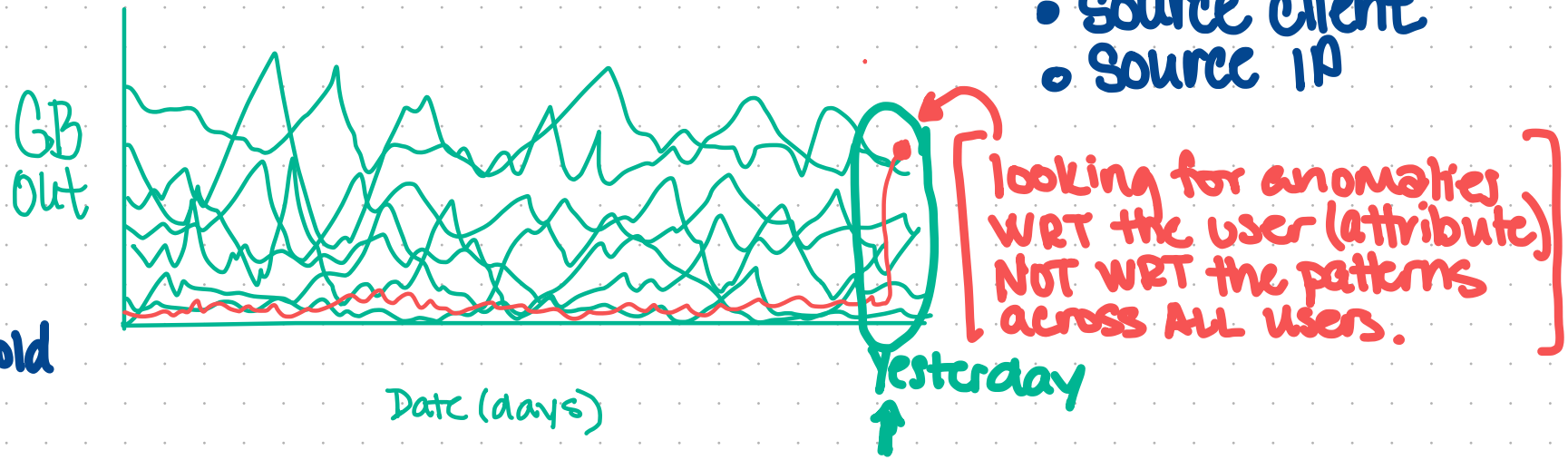
- daily
- Work hours
- Non-work hours
- Work days
- Non-work days

*by Attribute

- network user ID
- Sender email
- recipient email
- destination IP
- destination domain
- destination TLD
- Source client
- Source IP

GOAL:

A Model that will run daily, detect anomalies for the previous 24 hrs, an send alert to analyst when threshold crossed.



* by measuring in different ways, we identify different behaviors. e.g. maybe it is expected to see a 1 GB video go out but its not expected to see 50 uploads (that may total 1 GB.) 50 wouldnt be caught w/ 1st metric).

+ By modeling different time periods, we may see abnormal behavior in one that we couldn't see in another. e.g. ANY of data out over the day may not be abnormal, but when it all left @ 2:00 AM THAT'S an anomaly!

A threat actor may use the same user acct to authenticate in network to get data out, may use the same destination, the same source, the same client, etc. But will Rarely use all the same, and anomaly may not be visible in each of those attributes. So, by covering all, we can aggregate scores & have a "More" complete picture to help filter out noise.

WRANGLE:

- User ID
- total GB Out
- Date (Day)

→ Splunk -

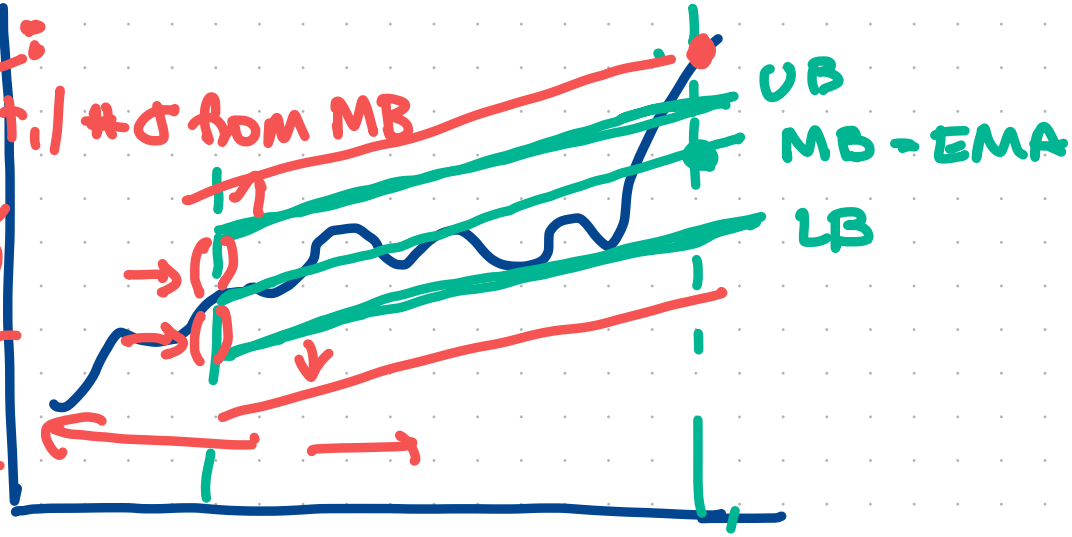
→
- User ID -
- Target Day - yesterday
- Current EMA -
- Midband
- Upper Band
- Lower Band
} Bollinger Bands
- Pct B -
- observation Window - 28days
- target value - GB out yesterday

→
- User ID
- Target Day
- % B
- outcome

≥1, Rank by %B desc.

Ways to tweak:

- increase the wt. / # σ from MB
- Adjusting the EMA window
- Adj. Smoothing - More/less wt. to more recent value.



%B > 1 ⇒ above UB

%B < 0 ⇒ below LB

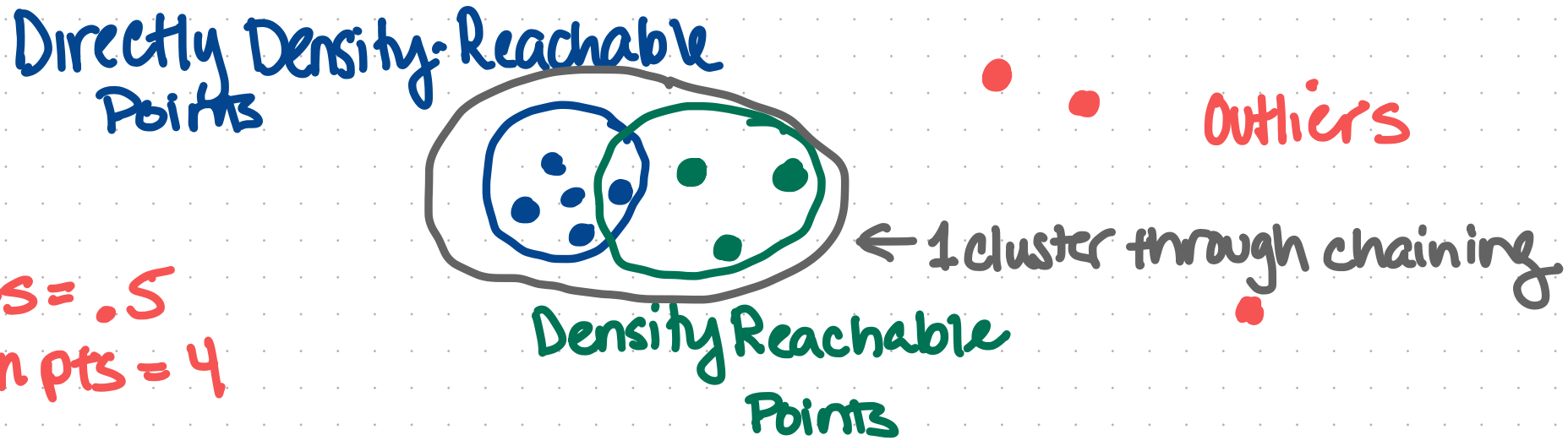
%B = .5 ⇒ EMA / MB

Finding Anomalies through Clustering -

DBSCAN : Density Based Spatial Clustering of Applications with Noise

EPS - epsilon - how close the points should be to each other to be considered part of a cluster

MinPts - min # of pts to form a dense region



eps = .5

Min pts = 4