



Defending Networks with Incomplete Information: A Machine Learning Approach

Alexandre Pinto
alexcp@mlsecproject.org
@alexcpsec
@MLSecProject



WARNING!

- This is a talk about DEFENDING not attacking
 - NO systems were harmed on the development of this talk.
 - This is NOT about some vanity hack that will be patched tomorrow
 - We are actually trying to BUILD something here.
- This talk includes more MATH than the daily recommended intake by the FDA.
- You have been warned...

Who's this guy?

- 12 years in Information Security, done a little bit of everything.
- Past 7 or so years leading security consultancy and monitoring teams in Brazil, London and the US.
 - If there is any way a SIEM can hurt you, it did to me.
- Researching machine learning and data science in general for the past year or so. Participates in Kaggle machine learning competitions (for fun, not for profit).
- First presentation at BlackHat! Thanks for attending!

Agenda

- Security Monitoring: We are doing it wrong
- Machine Learning and the Robot Uprising
- Data gathering for InfoSec
- Case study: Model to detect malicious activity from log data
- MLSec Project
- Attacks and Adversaries
- Future Direction

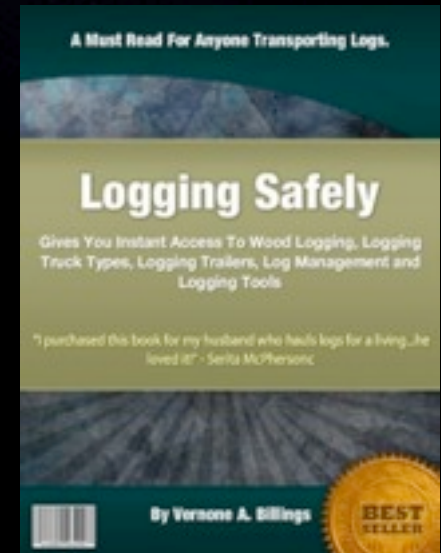
The Monitoring Problem

- Logs, logs everywhere



The Monitoring Problem

- Logs, logs everywhere



Are these the right tools for the job?



- SANS Eighth Annual 2012 Log and Event Management Survey Results (http://www.sans.org/reading_room/analysts_program/SortingThruNoise.pdf)

Are these the right tools for the job?



- SANS Eighth Annual 2012 Log and Event Management Survey Results (http://www.sans.org/reading_room/analysts_program/SortingThruNoise.pdf)

Correlation Rules: a Primer

- Rules in a SIEM solution invariably are:
 - “Something” has happened “x” times;
 - “Something” has happened and other “something2” has happened, with some relationship (time, same fields, etc) between them.
- Configuring SIEM = iterate on combinations until:
 - Customer or management is ~~foiled~~ satisfied; or
 - Consulting money runs out
- Behavioral rules (anomaly detection) helps a bit with the “x”s, but still, very laborious and time consuming.

Not exclusively a tool problem

- However, there are individuals who will do a good job
- How many do you know?
- DAM hard (ouch!) to find these capable professionals

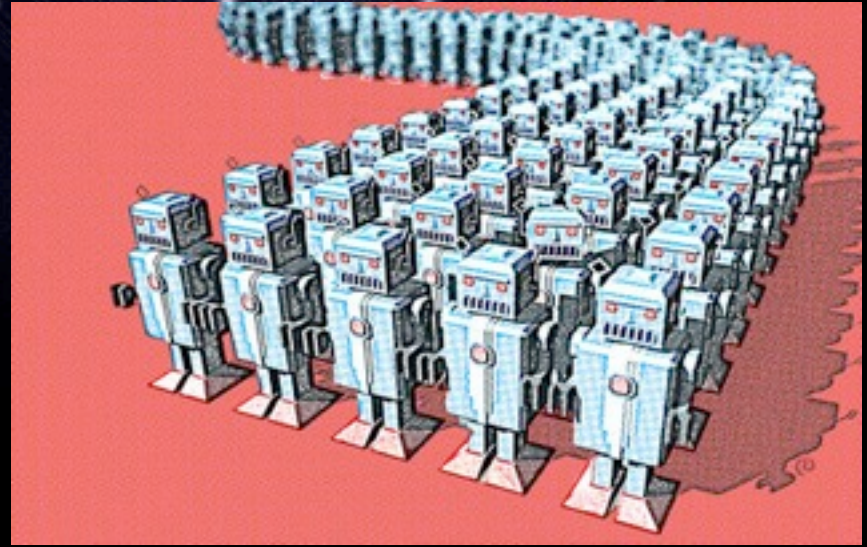


Next up: Big Data Technologies

- How many of these very qualified professionals will we need?
- How many know/ will learn statistics, data analysis, data science?



We need an Army! Of ROBOTS!



Enter Machine Learning

- “Machine learning systems automatically learn programs from data” (*)
- You don’t really code the program, but it is inferred from data.
- Intuition of trying to mimic the way the brain learns: that’s where terms like artificial intelligence come from.



(*) CACM 55(10) – A Few Useful Things to Know about Machine Learning

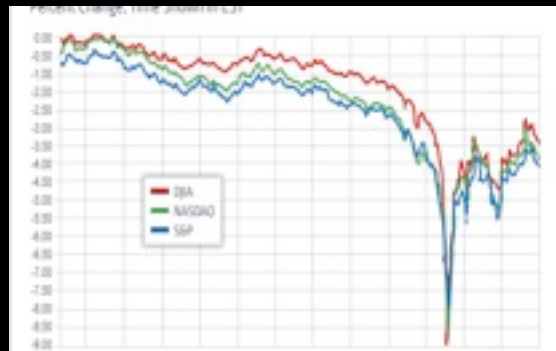
Applications of Machine Learning

- Sales



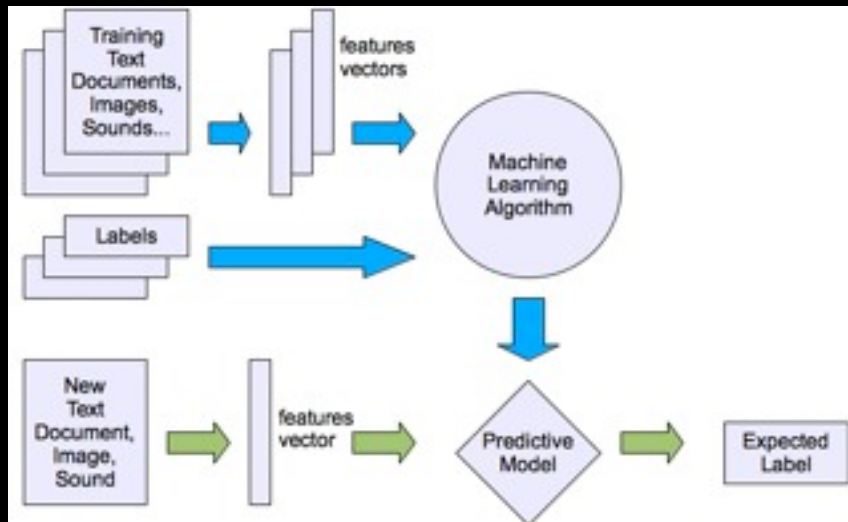
- Image and Voice Recognition

- Trading

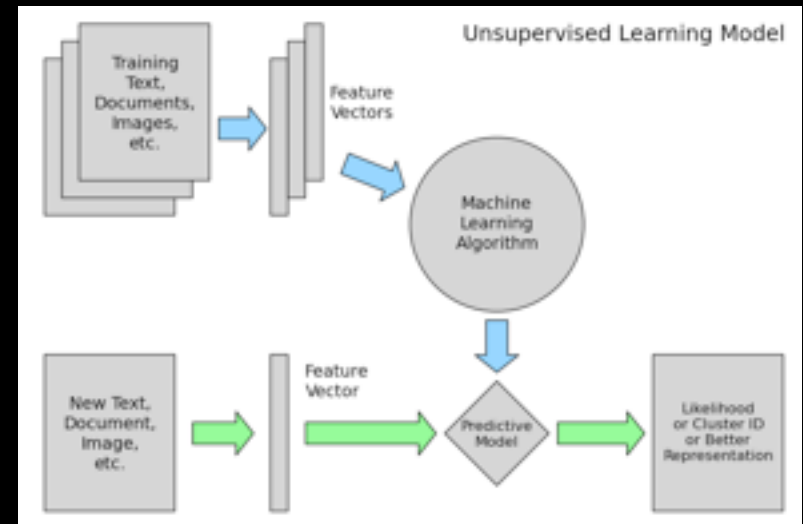


Kinds of Machine Learning

- Supervised Learning:
 - Classification (NN, SVM, Naïve Bayes)
 - Regression (linear, logistic)

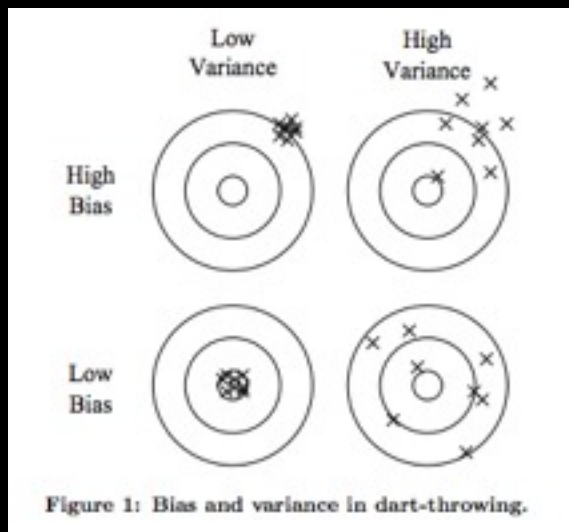


- Unsupervised Learning :
 - Clustering (k-means)
 - Decomposition (PCA, SVD)

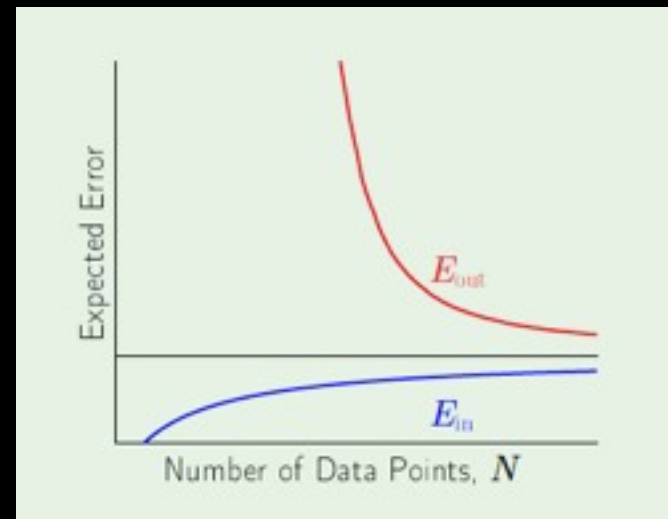


Considerations on Data Gathering

- Models will (generally) get better with more data
 - But we always have to consider bias and variance as we select our data points
 - Also adversaries – we may be force-fed “bad data”, find signal in weird noise or design bad (or exploitable) features
- “I’ve got 99 problems, but data ain’t one”



Domingos, 2012



Abu-Mostafa, Caltech, 2012

Considerations on Data Gathering

- Adversaries – Exploiting the learning process
- Understand the model, understand the machine, and you can circumvent it
- Something InfoSec community knows very well
- Any predictive model on Infosec will be pushed to the limit
- Again, think back on the way SPAM engines evolved.



Designing a model to detect external agents with malicious behavior

- We've got all that log data anyway, let's dig into it
- Most important (and time consuming) thing is the "feature engineering"
- We are going to go through one of the algorithms I have put together as part of my research



Model: Data Collection

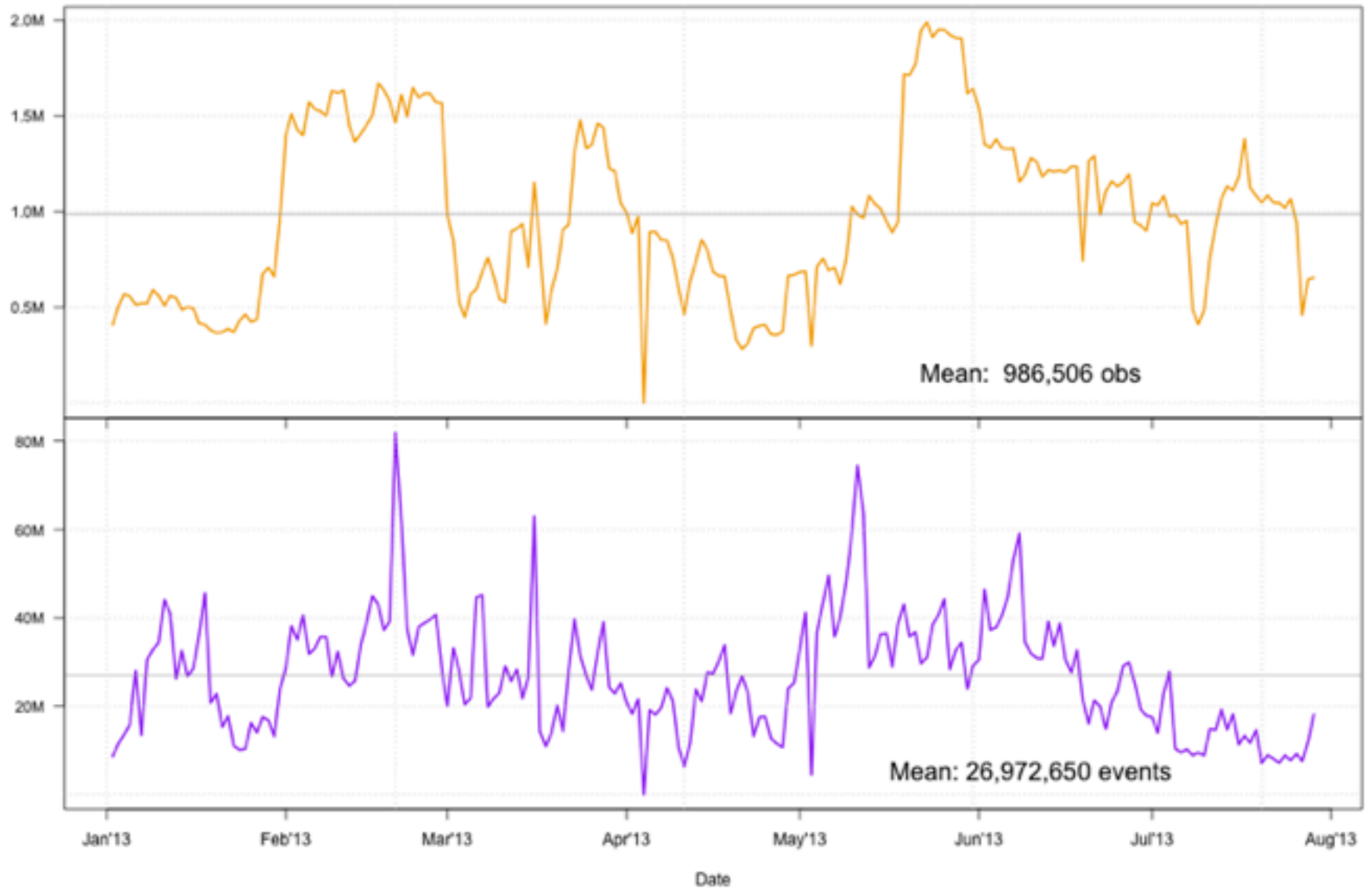
- Firewall block data from SANS DShield (per day)
- Firewalls, really? Yes, but could be anything.
- We get summarized “malicious” data per port

```
> sans
```

	date	ip	targetPort	protocol	reports	targets	firstSeen	lastSeen
1:	20130622	89.248.171.125	80	TCP	64853	64775	00:14:14	17:51:54
2:	20130622	93.174.93.179	80	TCP	59580	58487	05:11:15	22:21:41
3:	20130622	213.186.60.63	80	TCP	58429	58429	00:15:41	21:42:28
4:	20130622	202.121.166.203	22	TCP	106621	53328	05:18:26	10:10:33
5:	20130622	218.207.176.125	80	TCP	53241	53241	21:16:09	21:56:07

1107159:	20130622	65.55.37.104	16766	TCP	2	1	12:31:06	12:31:12
1107160:	20130622	65.55.37.104	16765	TCP	1	1	00:45:24	00:45:24
1107161:	20130622	65.55.37.104	16761	TCP	3	1	09:47:49	09:48:39
1107162:	20130622	65.55.37.104	16759	TCP	2	1	03:29:51	03:30:37
1107163:	20130622	65.55.37.104	16721	TCP	1	1	20:29:24	20:29:24

Number of Reports and Events per day



- Number of aggregated events (orange)
- Number of log entries before aggregation (purple)

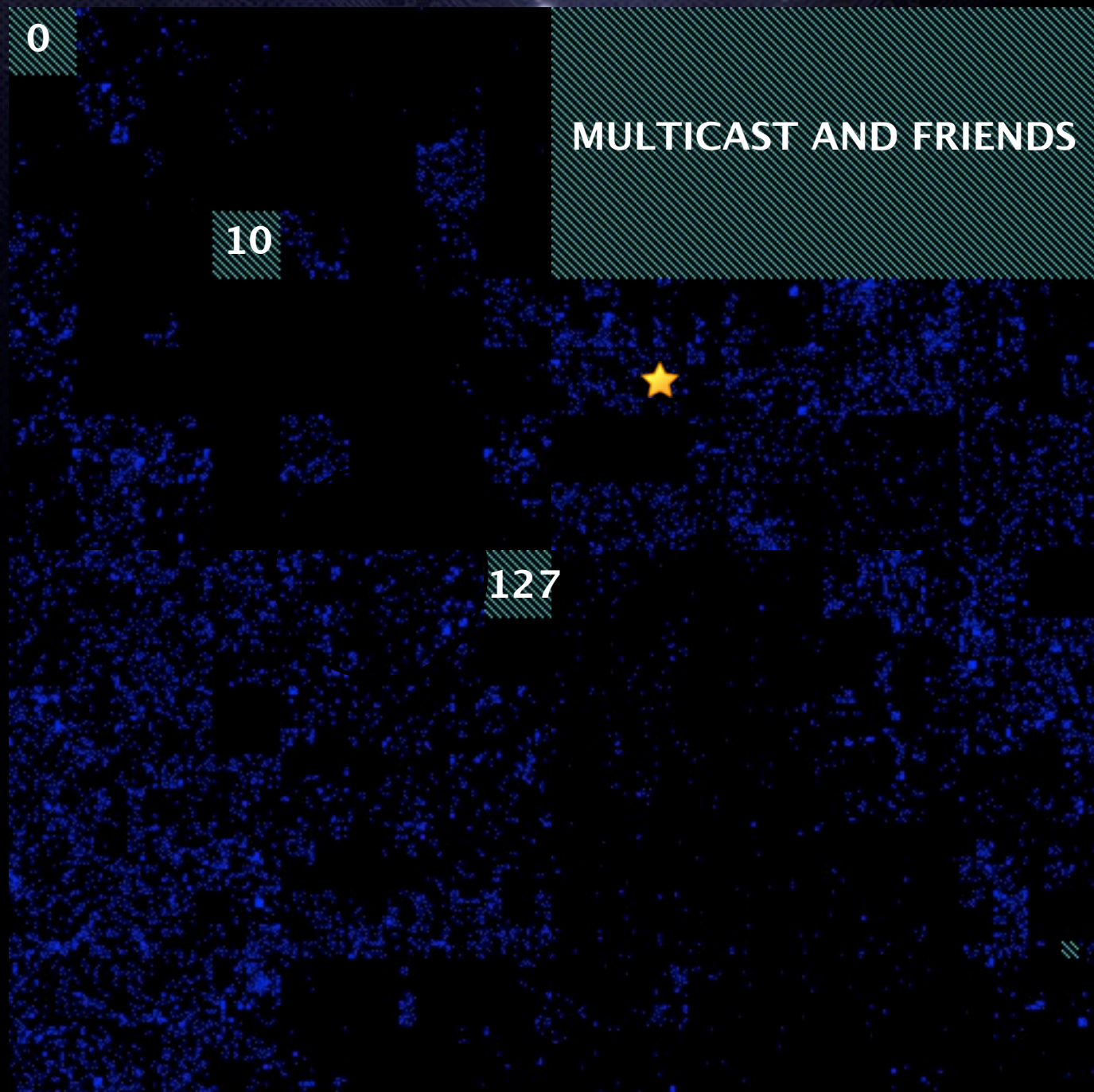
Model Intuition: Proximity

- Assumptions to aggregate the data
- Correlation / proximity / similarity BY BEHAVIOR
- “Bad Neighborhoods” concept:
 - Spamhaus x CyberBunker
 - Google Report (June 2013)
 - Moura 2013
- Group by Netblock (/16, /24)
- Group by ASN
 - (thanks, Team Cymru)



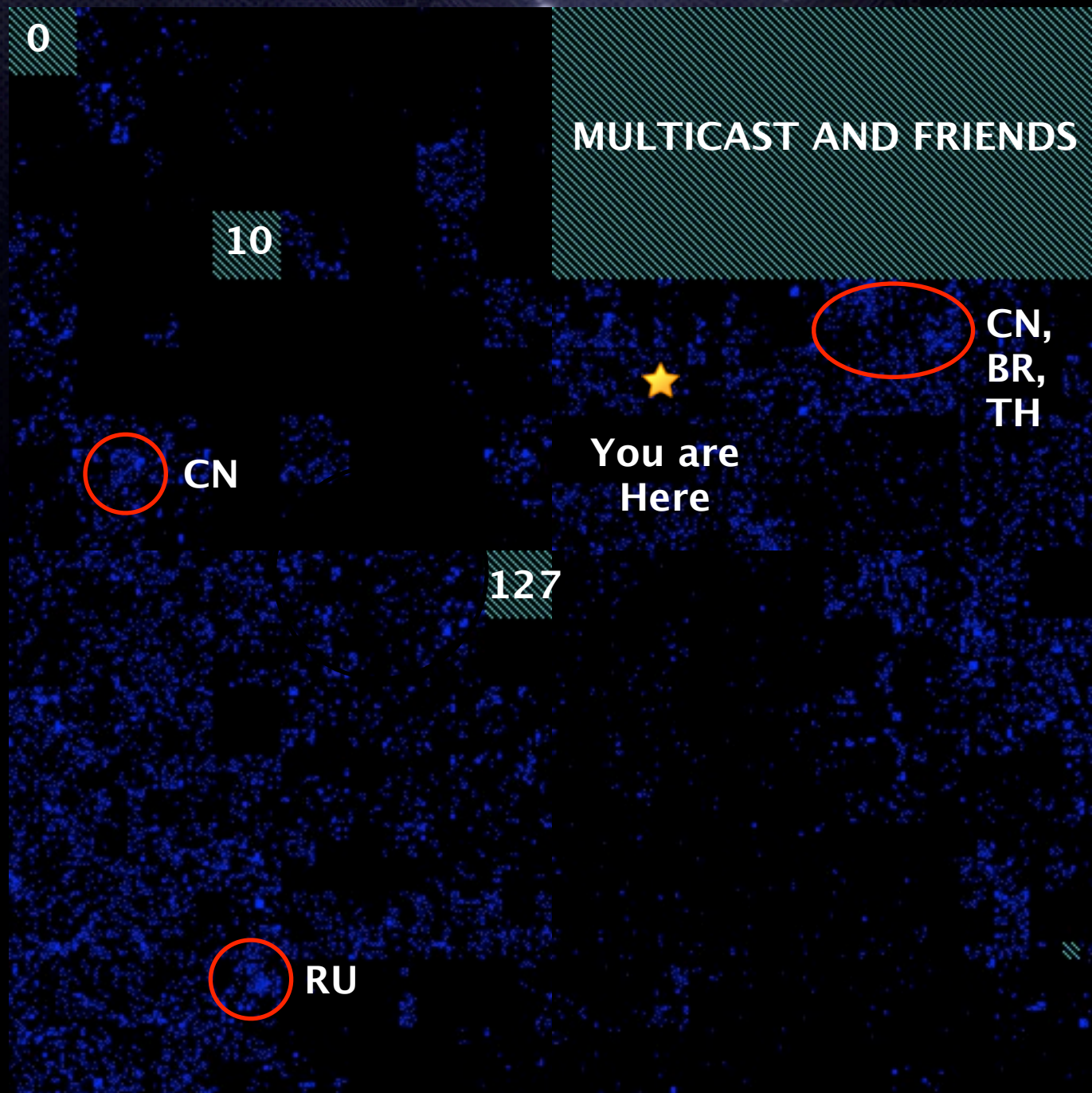
Map of the
Internet
(Hilbert Curve)
Block port 22
2013-07-20

Not random at
all...

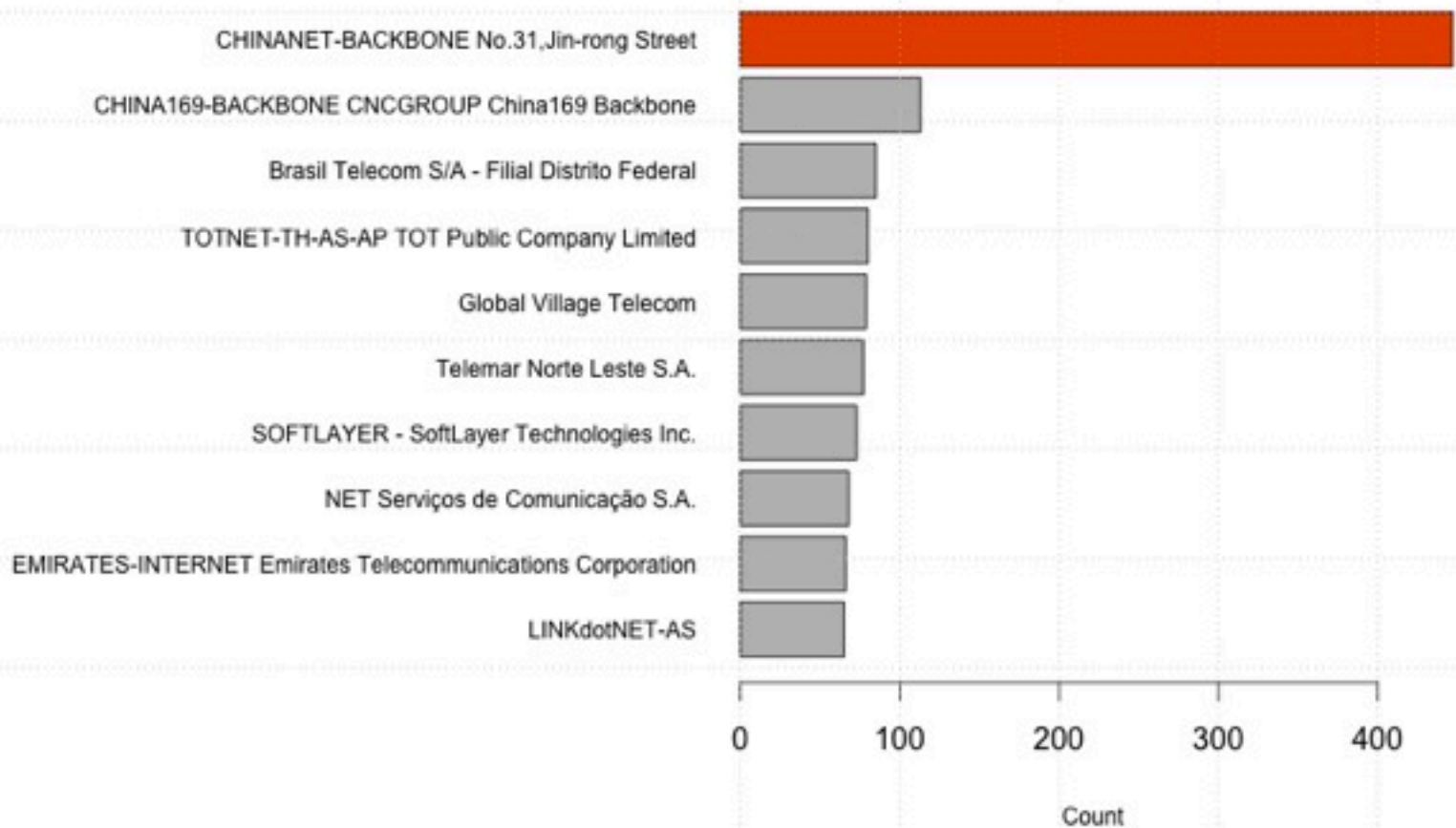


Map of the Internet
(Hilbert Curve)
Block port 22
2013-07-20

Not random at all...

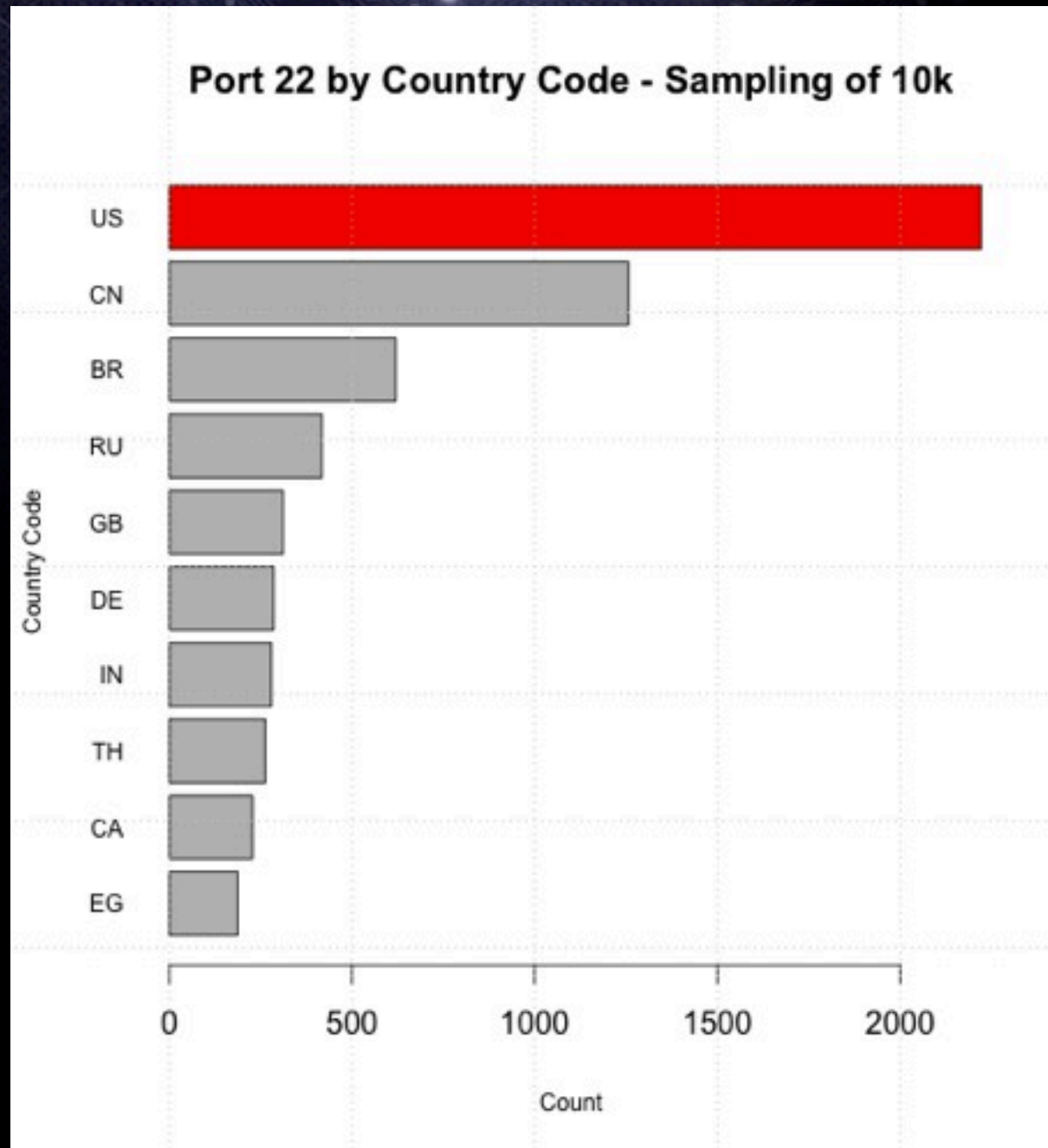


Port 22 by AS Name - Sampling of 10k



Be careful with confirmation bias

Country codes are not enough for any prediction power of consequence today



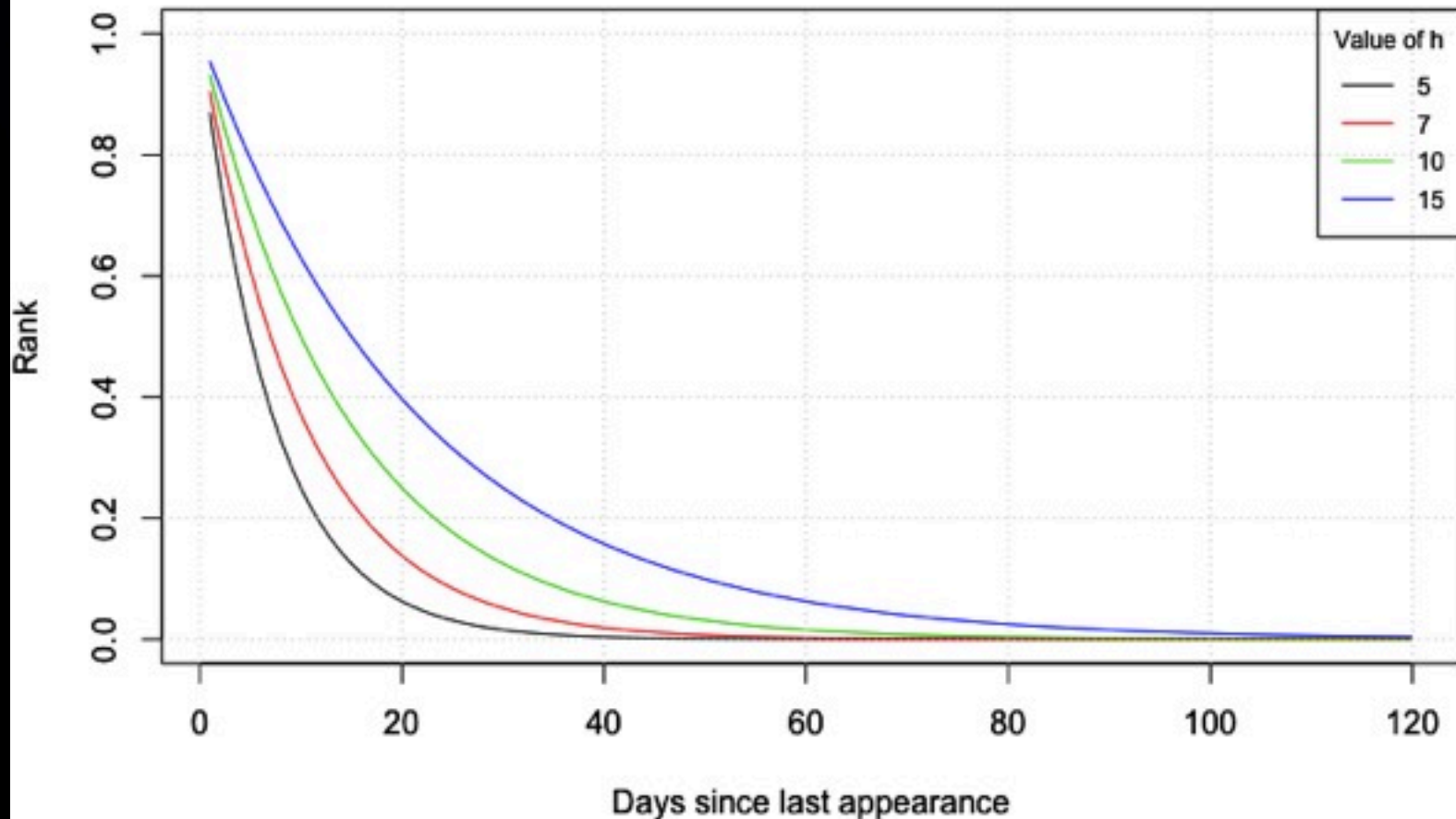
Model Intuition: Temporal Decay

- Even bad neighborhoods renovate:
 - Agents may change ISP, Botnets may be shut down
 - A little paranoia is Ok, but not EVERYONE is out to get you (at least not all at once)
- As days pass, let's forget, bit by bit, who attacked
- A Half-Life decay function will do just fine



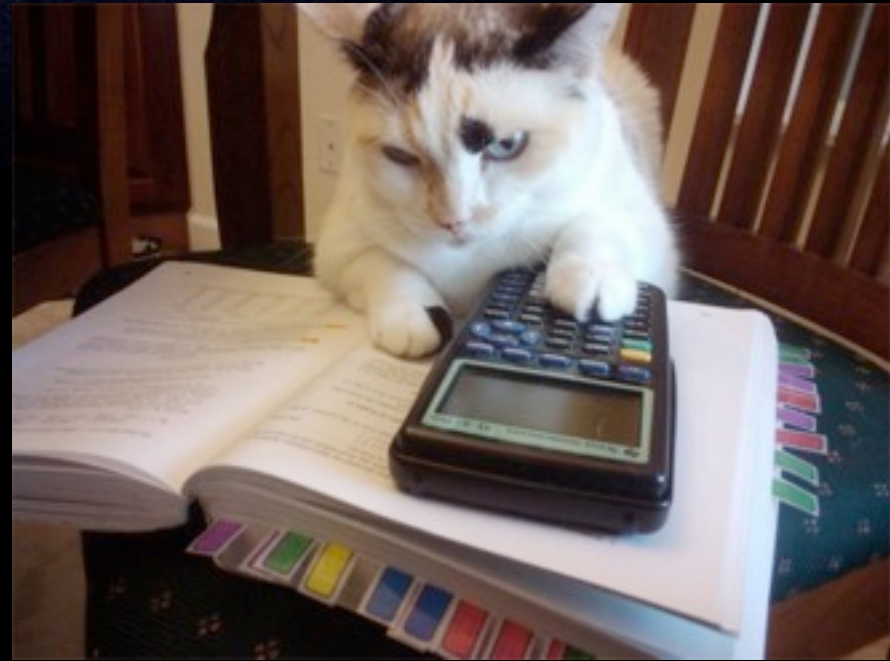
Model Intuition: Temporal Decay

Exponential Decay per Half-life



Model: Calculate Features

- Cluster your data: what behavior are you trying to predict?
- Create “Badness” Rank = lwRank (just because)
- Calculate normalized ranks by IP, Netblock (16, 24) and ASN
- Missing ASNs and Bogons (we still have those) handled separately, get higher ranks.

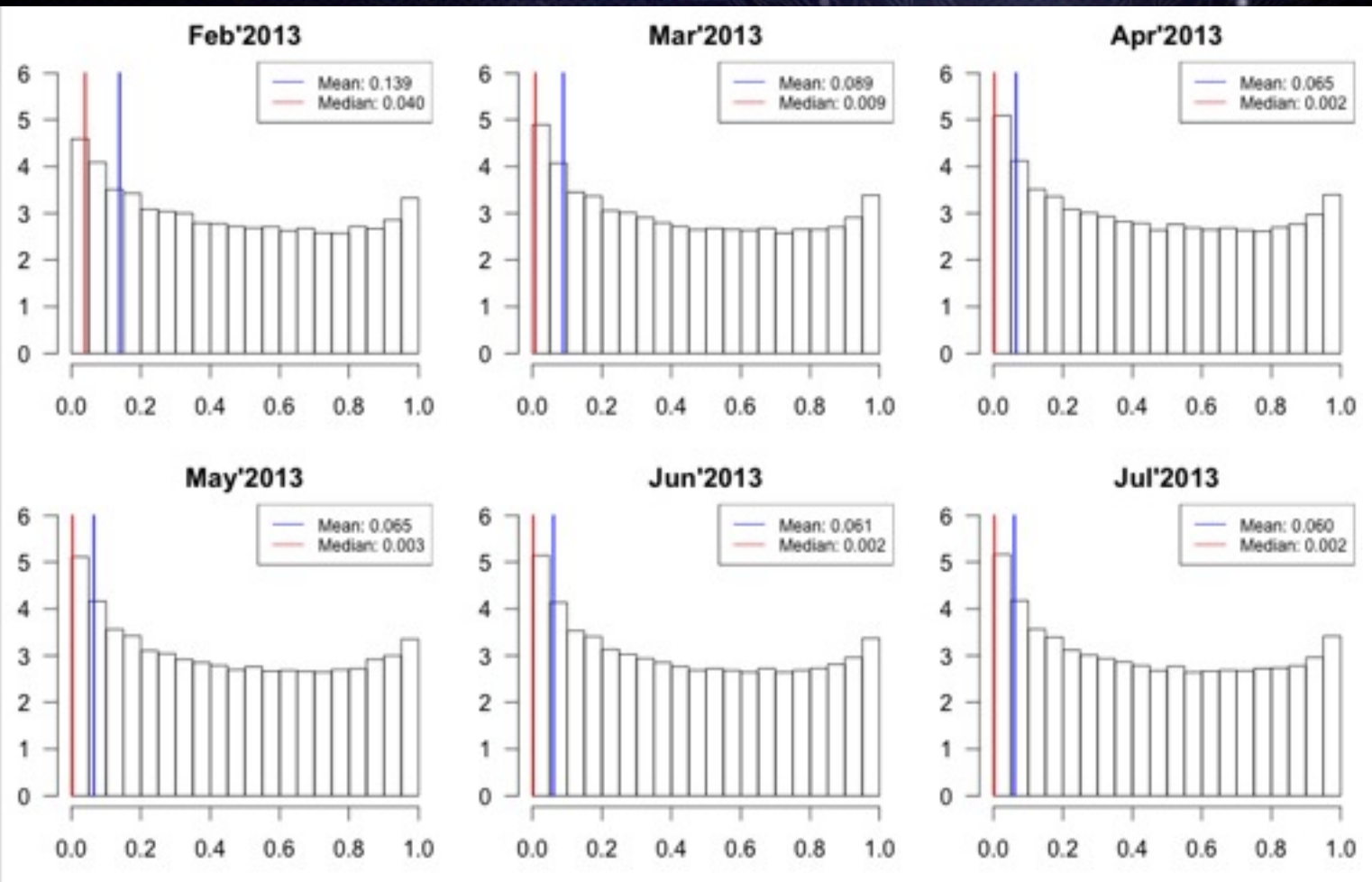


Model: Calculate Features

- We will have a rank calculation per day:
 - Each “day-rank” will accumulate all the knowledge we gathered on that IP, Netblock and ASN to that day
 - Decay previous “day-rank” and add today’s results
- Training data usually spans multiple days
- Each entry will have its date:
 - Use that “day-rank”
 - NO cheating ----->
 - Survivorship bias issues!

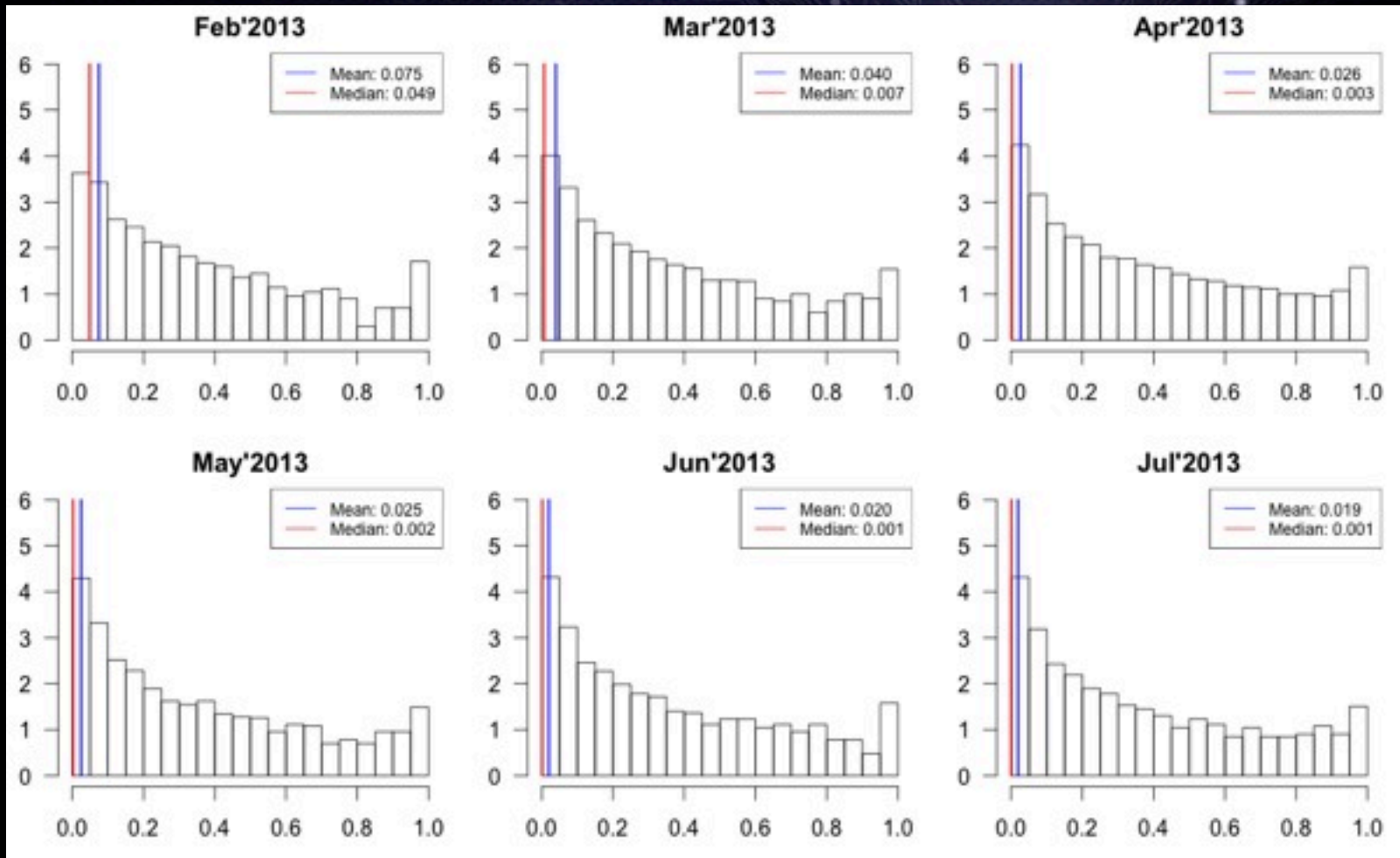


Model: Example Feature (1)



- Block on Port 3389 (IP address only)
 - Horizontal axis: lwRank from 0 (good/neutral) to 1 (very bad)
 - Vertical axis: $\log_{10}(\text{number of IPs in model})$

Model: Example Feature (2)



- Block on Port 22 (IP address only)
 - Horizontal axis: lwRank from 0 (good/neutral) to 1 (very bad)
 - Vertical axis: $\log_{10}(\text{number of IPs in model})$

Training the Model

- YAY! We have a bunch of numbers per IP address!
- We get the latest blocked log files (SANS or not):
 - We have “badness” data on IP Addresses – features
 - If they were blocked, they are “malicious” – label
- Now, for each behavior to predict:
 - Create a dataset with “enough” observations:
 - Rule of Thumb: 70k – 120k is good because of empirical dimensionality.

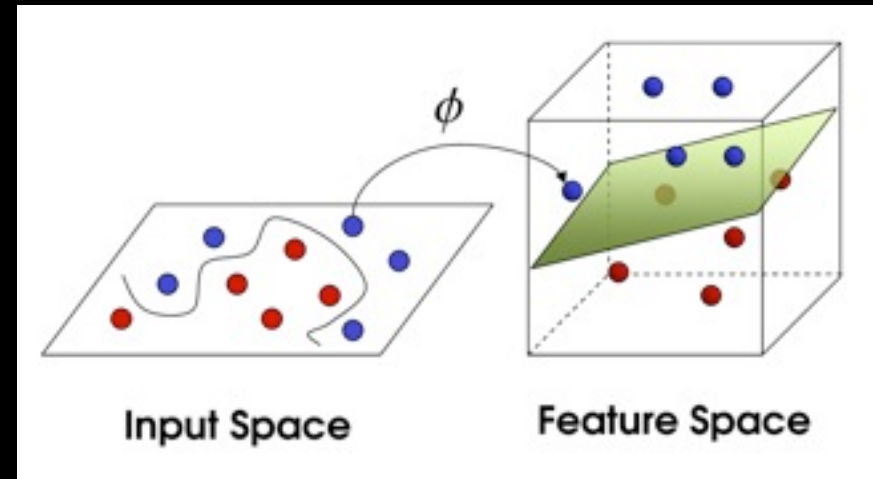
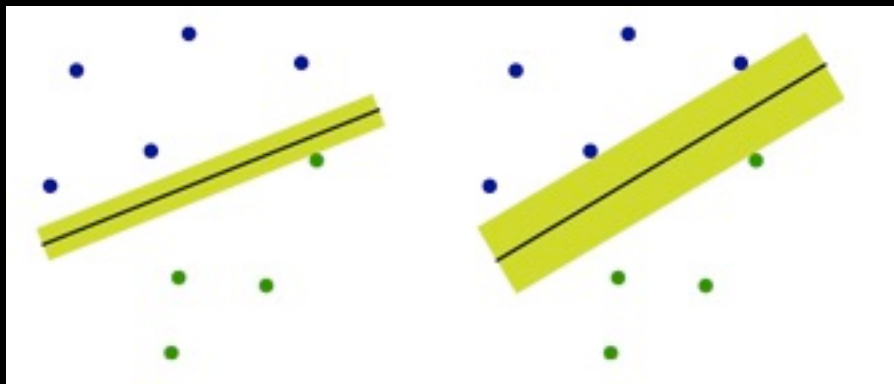
Negative and Positive Observations

- We also require “non-malicious” IPs!
- If we just feed the algorithms with one label, they will get lazy.
- CHEAP TRICK: Everything is “malicious” – trivial solution
- Gather “non-malicious” IP addresses from Alexa and Chromium Top 1m Sites.



SVM FTW!

- Use your favorite algorithm! YMMV.
- I chose Support Vector Machines (SVM):
 - Good for classification problems with numeric features
 - Not a lot of features, so it helps control overfitting, built in regularization in the model, usually robust
 - Also awesome: hyperplane separation on an unknown infinite dimension.



Jesse Johnson – shapeofdata.wordpress.com

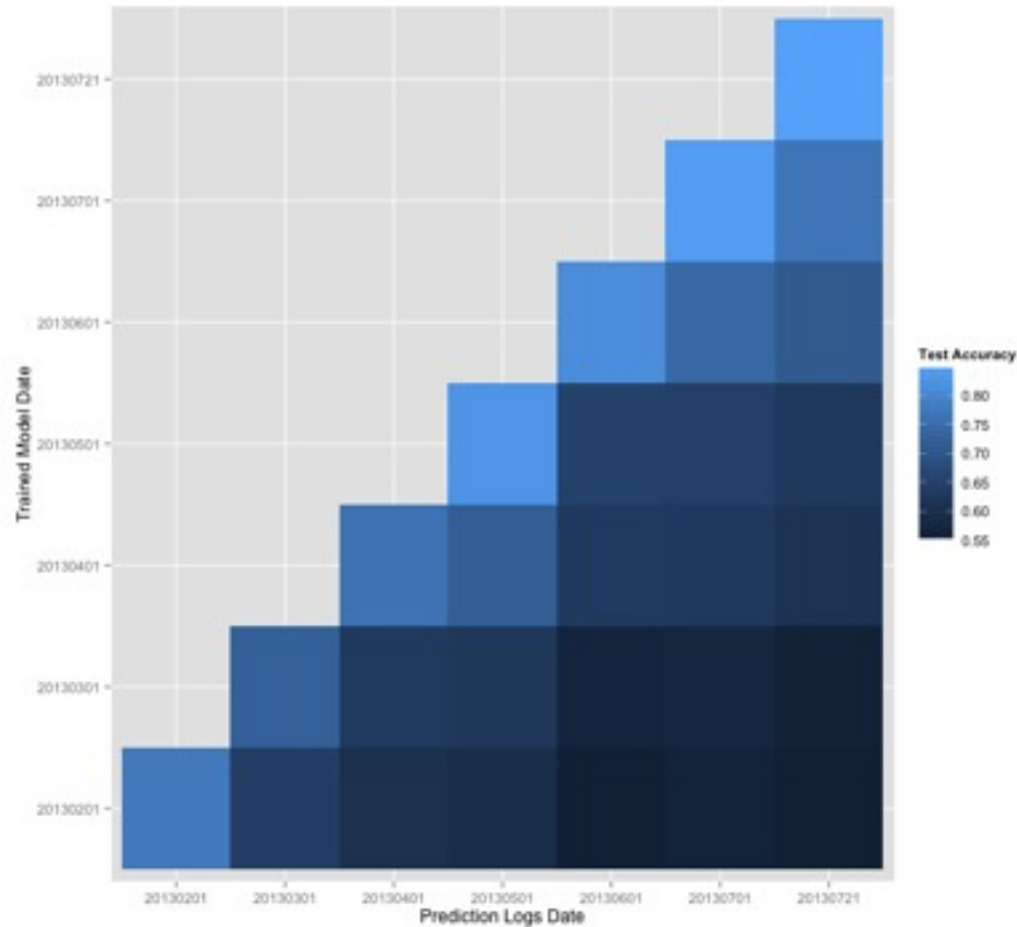
Results: Training/Test Data

- Model is trained on each behavior for each day
- Training accuracy* (cross-validation): 83 to 95%
- New data – test accuracy*:
 - Training model on day D, predicting behavior in day D+1
 - 79 to 95%, roughly increasing over time

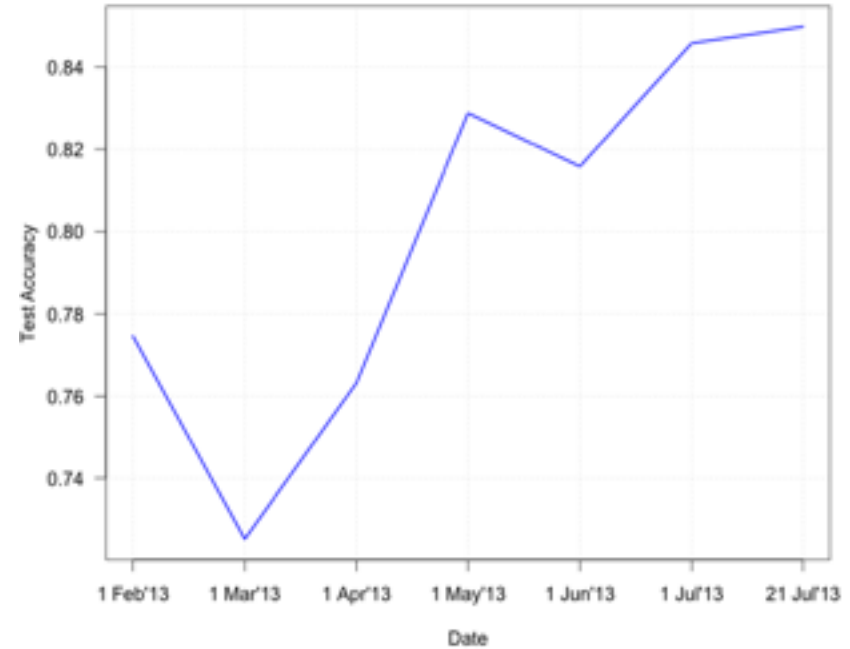
(*)Accuracy = (things we got right) / (everything we tried)

Results: Training/Test Data

Model Test Accuracy - Port 25

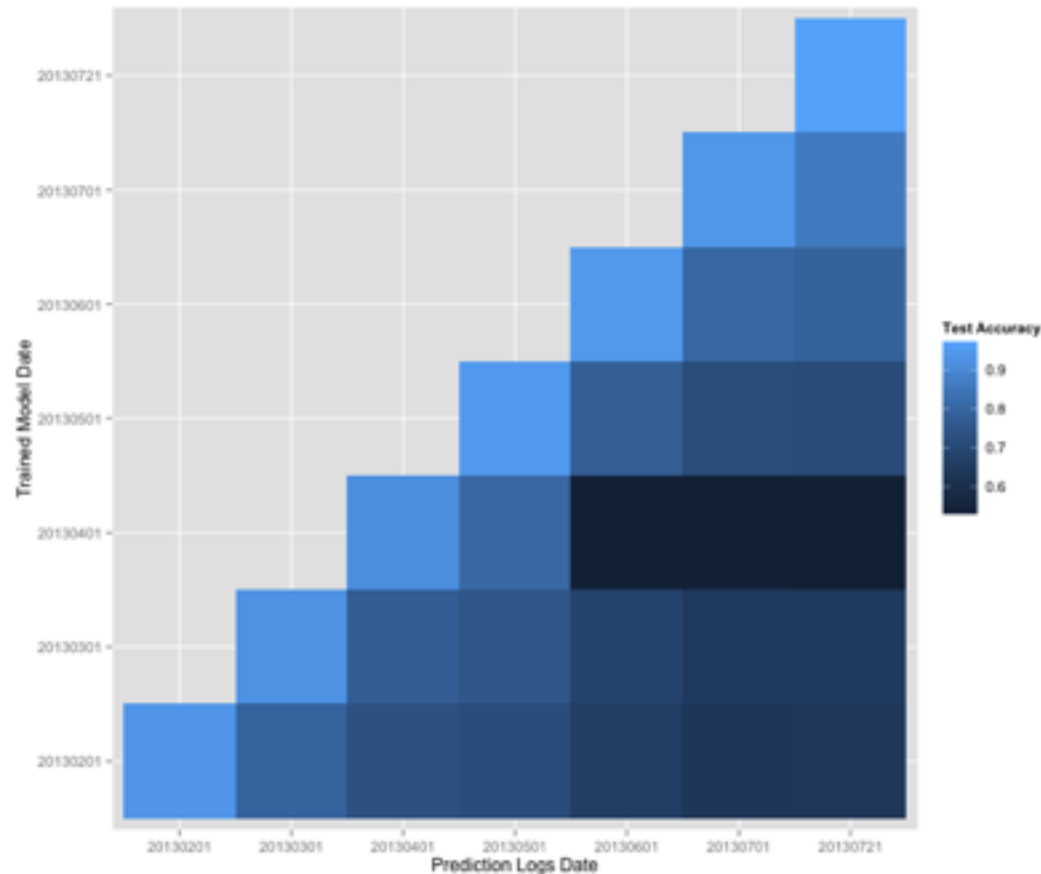


Test Accuracy for Port 25 (Same day)

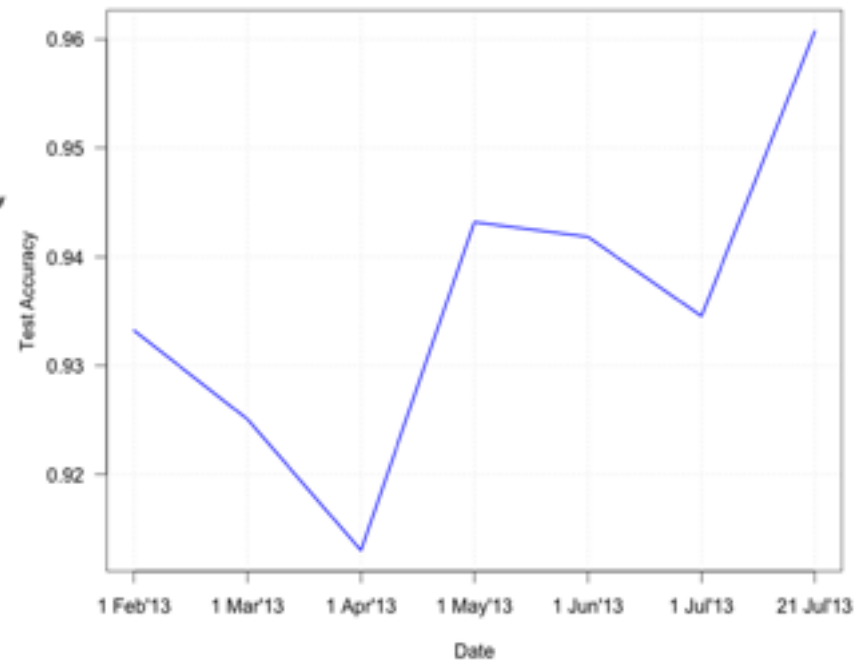


Results: Training/Test Data

Model Test Accuracy - Port 3389



Test Accuracy for Port 3389 (Same day)



Results: New Data

$$LR_+ = \frac{\Pr(T+|D+)}{\Pr(T+|D-)}$$

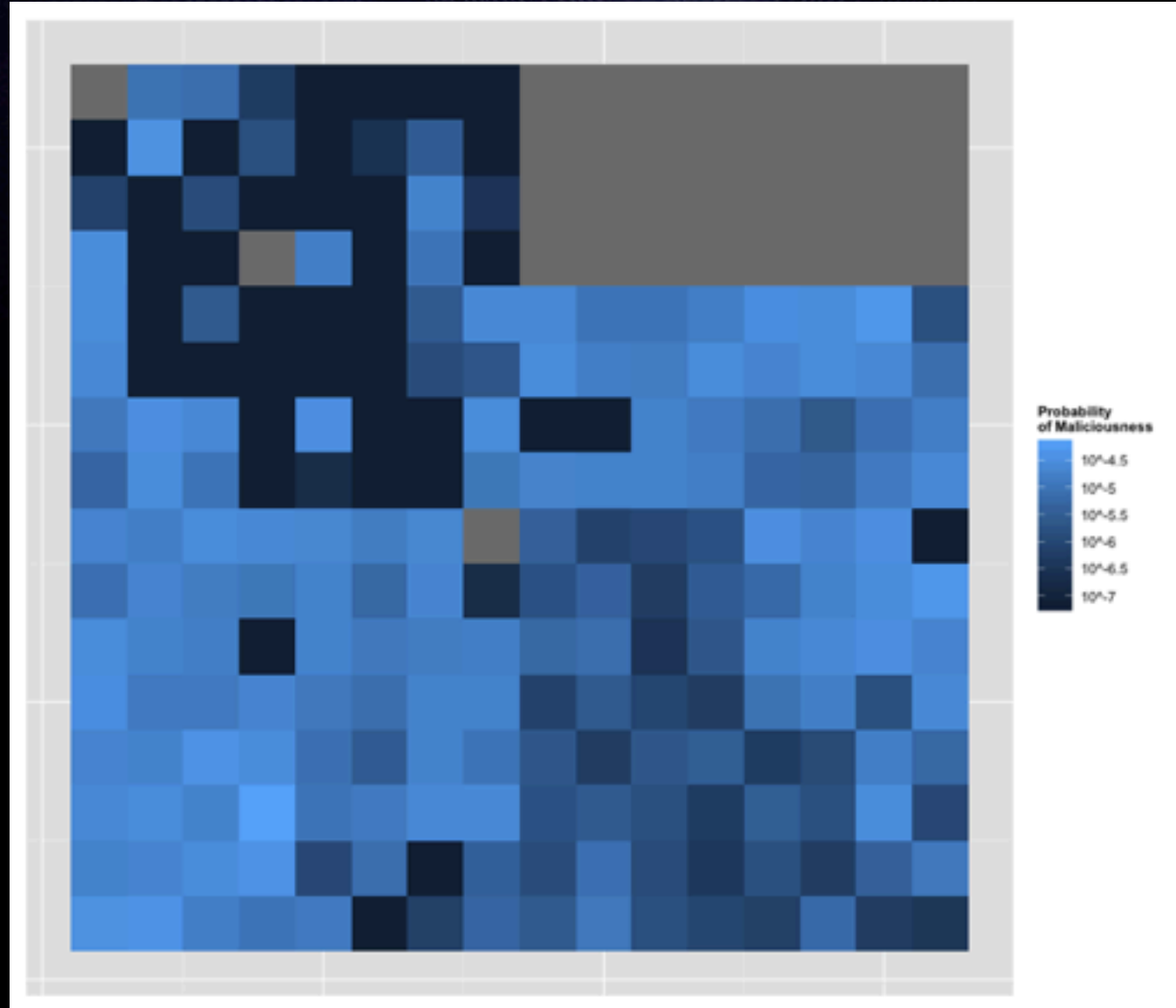
- How does that help?
- With new data we can verify the labels, we find:
 - 70 – 92% true positive rate (sensitivity/precision)
 - 95 – 99% true negative rate (specificity/recall)
- This means that (odds likelihood calculation):
 - If the model says something is “bad”, it is 13.6 to 18.5 times MORE LIKELY to be bad.
- Think about this.
- Wouldn't you rather have your analysts look at these first?

Remember the Hilbert Curve?

Behavior: block
on port 22

Trial inference
on 100k IP
addresses per
Class A subnet

Logarithm
scale:
brightest tiles
are 10 to 1000
times more
likely to
attack.

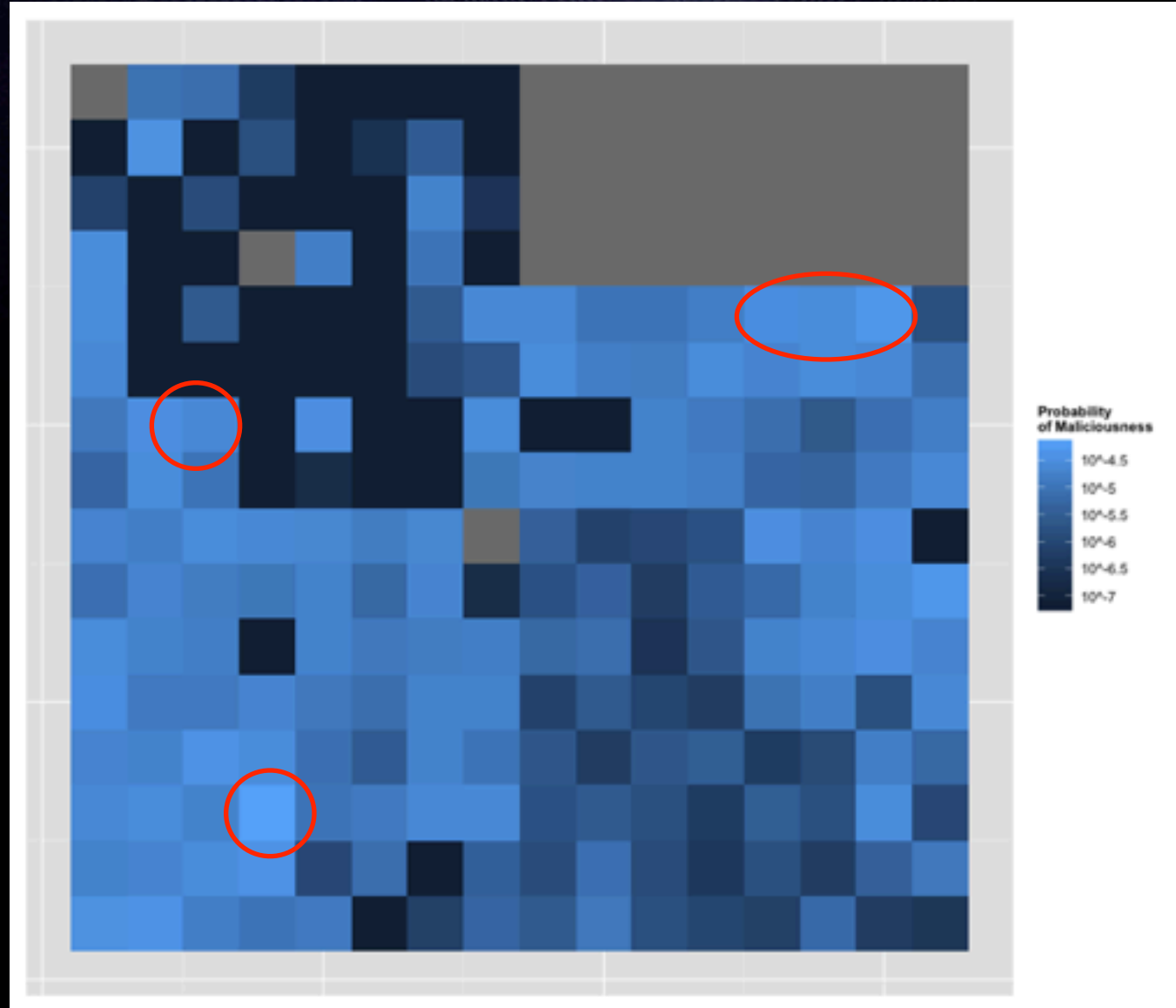


Remember the Hilbert Curve?

Behavior: block
on port 22

Trial inference
on 100k IP
addresses per
Class A subnet

Logarithm
scale:
brightest tiles
are 10 to 1000
times more
likely to
attack.



Attacks and Adversaries

- IP addresses are not as reliable as they could be:
 - Forget about UDP
 - Lowest possible value for DFIR
- This is not attribution, this is defense
- Challenges:
 - Anonymous proxies (not really, same rules apply)
 - Tor (less clustering behavior on exit nodes)
 - Fast-flux Tor – 15~30 mins
- Process was designed with different actors in mind as well, given they can be clustered in some way.

Future Direction

- As is, the results from the predictions can help Security Analysts on tiers 1 and 2 of SOCs:
 - You can't "eyeball" all of the data.
 - Makes the deluge of logs produce something actionable
- The real kicker is when we compose algorithms (ensemble):
 - Web server -> go through firewall, then IPS, then WAF
 - increased precision by composing different behaviors
- Given enough predictive power (increased likelihood):
 - Implement an SDN system that sends detected attackers through a "longer path" or to a Honeynet
 - Connection could be blocked immediately

Final Remarks

- Sign up, send logs, receive reports generated by models!
 - FREE! I need the data! Please help! ;)
- Looking for contributors, ideas, skeptics to support project as well.
- Please visit <https://www.mlsecproject.org> , message @MLSecProject or just e-mail me.



Take Aways



- Machine learning can assist monitoring teams in data-intensive activities (like SIEM and security tool monitoring)
- The odds likelihood ratio (12x to 18x) is proportional to the gain in efficiency on the monitoring teams.
- This is just the beginning! Lots of potential!
- MLSec Project is cool, check it out and sign up

Thanks!

- Q&A?
- Don't forget to submit feedback!

Alexandre Pinto
alexcp@mlsecproject.org
@alexcpsec
@MLSecProject



"Prediction is very difficult, especially if it's about the future."

– Niels Bohr