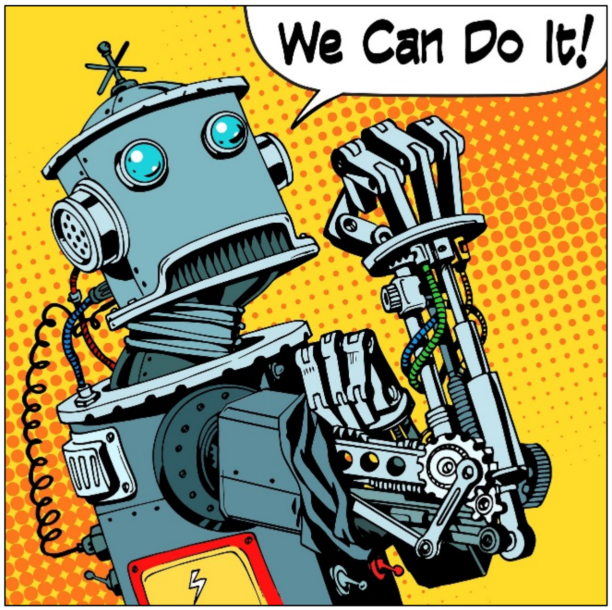
Applied Machine Learning in Software Security

Eliezer Kanal

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

What is Machine Learning?



Tom Mitchell, former CMU Machine Learning department chair:

> *The field of Machine Learning asks the question, "How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?"*

Machine Learning seeks to automate data analysis and inference.

What is Machine Learning?

If your problem can be stated as either of the following:

I would like to use ____ data to guess what ____ is.

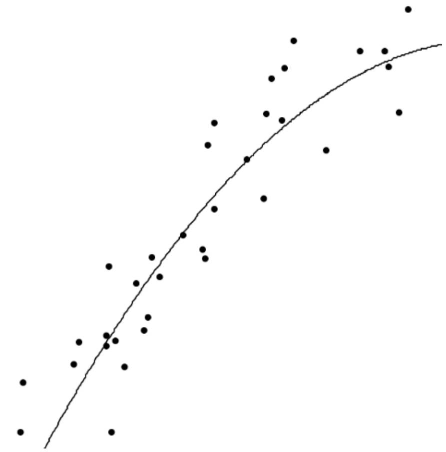
I would like to use _____data to predict _____.

…you would likely benefit from machine learning.

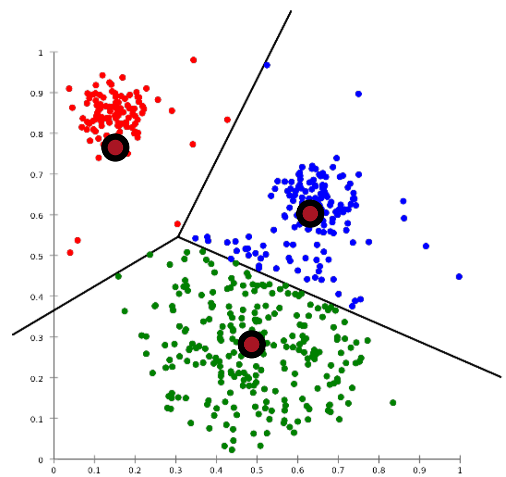
What is Machine Learning?

Sample Techniques:

- Regression



- K-Means Clustering



Clustering image: Weston.pace, https://commons.wikimedia.org/wiki/File:K_Means_Example_Step_4.svg

What is Machine Learning?

Feature Engineering:

Using existing data to create **more informative** data

Data Types	
Image	Static Video
Time series	Financial data Event counts
Structured text	Web forms Structured data (JSON, XML) Source code
Free text	News Tweets Email
many more…	

What is Machine Learning?

Examples:

- I would like to use **incident ticket** data to predict **customer needs**.
- I would like to use **publicly available code** to predict **what code I will write**.
- I would like to use **bug report** data to guess the location of **undetected bugs in my code**.

Autocomplete from Stack Over ×

Secure | https://emilschutte.com/stackoverflow-autocomplete/

Autocomplete from Stack Overflow

by Emil Schutte

Tired of writing code? Me too! Let's have Stack Overflow do it.

```
// Boss wants this function done by tomorrow :(
function contains(needle, haystack) {
  var

```

(Try typing a space. JavaScript only, for now.)

How it works

I grabbed a Stack Overflow data dump from https://archive.org/details/stackexchange and scraped out any code snippets from

- accepted answers
- with more than 50 points
- on posts tagged "javascript"

Then I processed it by walking the ASTs of those snippets and creating a "completion" fragment for each node, pairing a trace of the left-hand context with the code snippet for the right-hand side.

To complete at run time, it uses the same logic to find the left-hand trace at the current cursor position, and tries to match that up against the database of completion fragments. Available completions are sorted by a proprietary blend of post score, left-hand context similarity, and nearby identifiers.

What is Machine Learning?

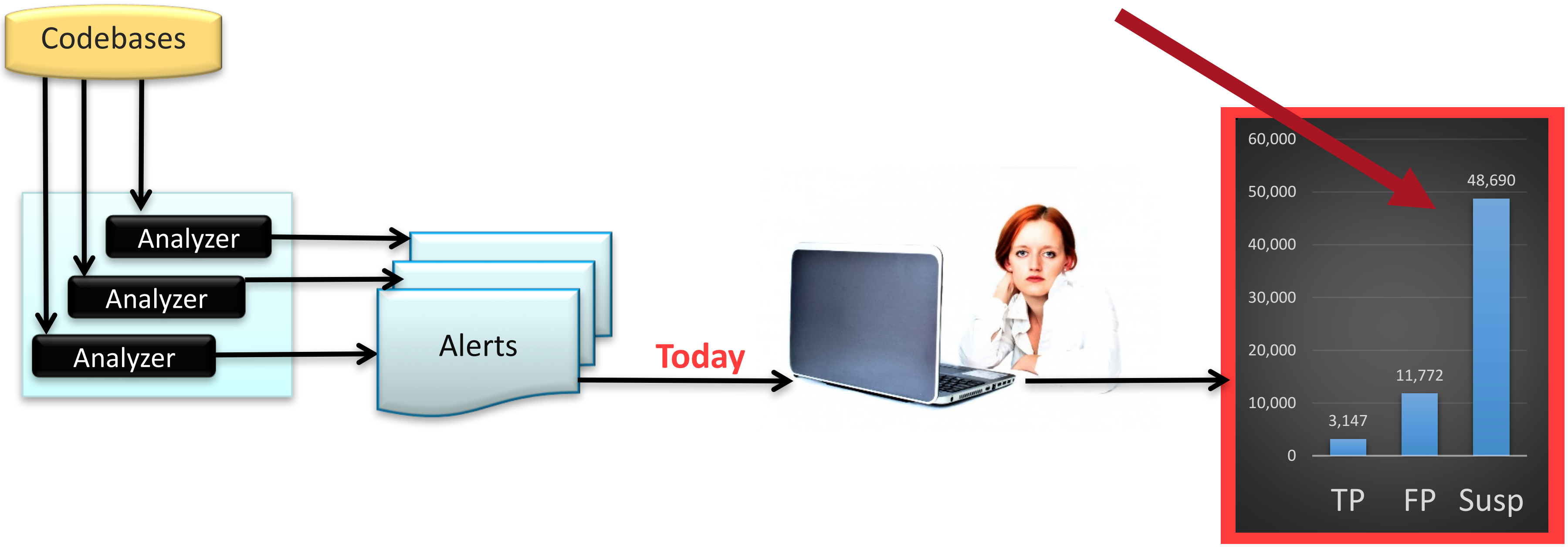
Examples:

- I would like to use **incident ticket** data to predict **customer needs**.
- I would like to use **publicly available code** to predict **what code I will write**.
- I would like to use **bug report** data to guess the location of **undetected bugs in my code**.

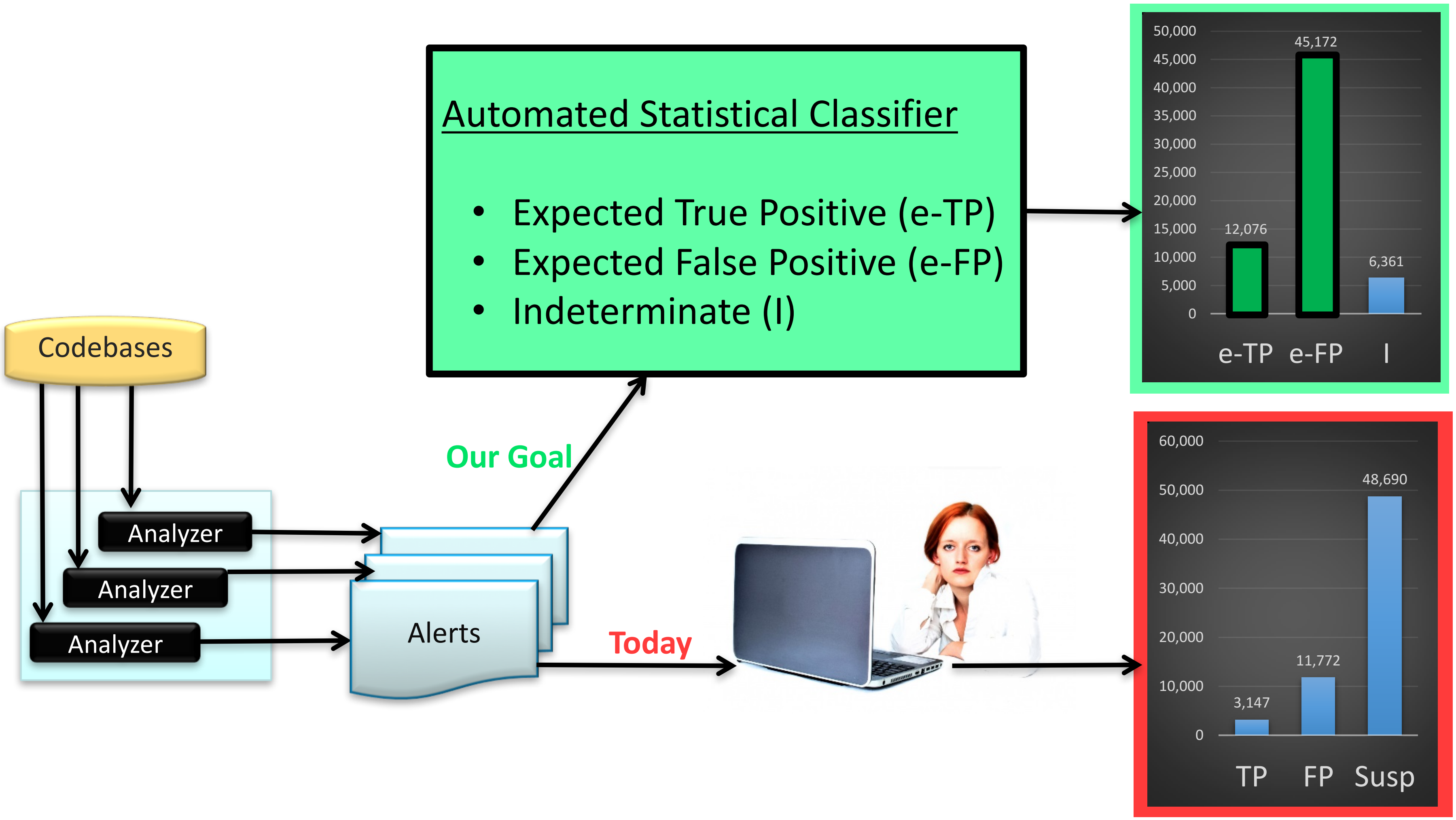
Applied ML: Vulnerability Detection

Applied ML: Vulnerability Detection

Many alerts left unaudited!



Applied ML: Vulnerability Detection



Applied ML: Vulnerability Detection

Classifiers
Lasso Logistic Regression
CART
Random Forest
Extreme Gradient Boosting (XGBoost)

Some of the features used	
Analysis tools used	Tokens in func/method
Significant LOC	Alerts in func/method
Complexity	Alerts in file
Coupling	Methods in file
Cohesion	SLOC in file
SEI coding rule	Avg Tokens
Function/method length	Avg SLOC
SLOC in func/method	Depth in code repository
# parameters in func/meth.	Cyclomatic complexity (func/meth)

Applied ML: Vulnerability Detection

<u>Significant improvement!</u>

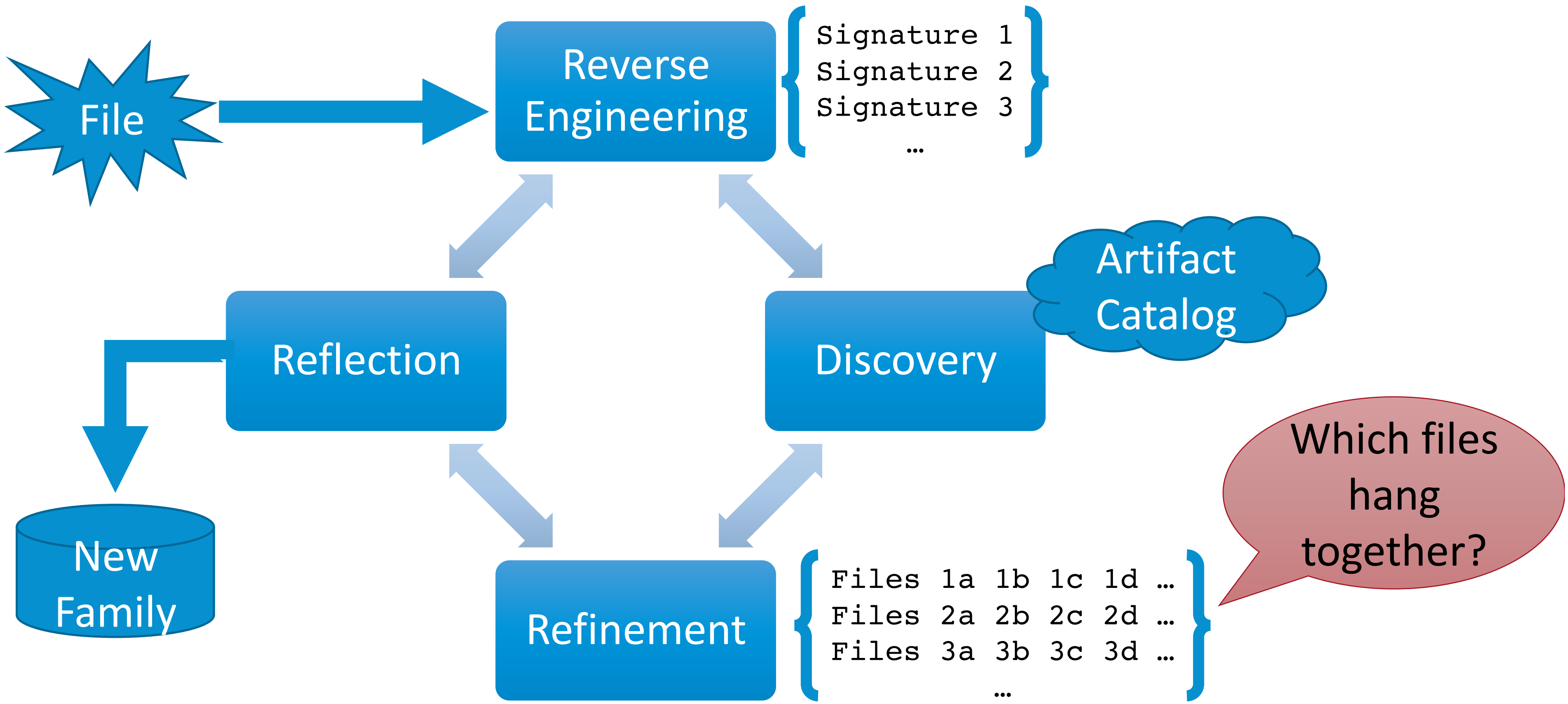
- 91% Classifier accuracy overall
- Specific rule accuracy at right
- 10x developer time saved!

Rule ID	XGBoost
INT31-C	97%
EXP01-J	74%
OBJ03-J	83%
FIO04-J*	80%
EXP33-C*	83%
EXP34-C*	72%
DCL36-C*	100%
ERR08-J*	100%
IDS00-J*	96%
ERR01-J*	100%
ERR09-J*	88%

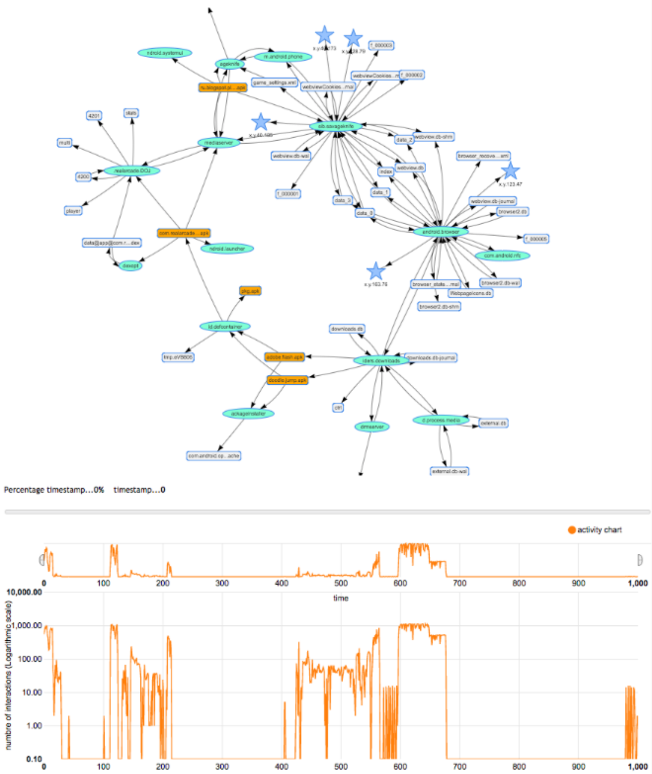
* Small quantity of data

Applied ML:
Malware family classification

Applied ML: Malware family classification

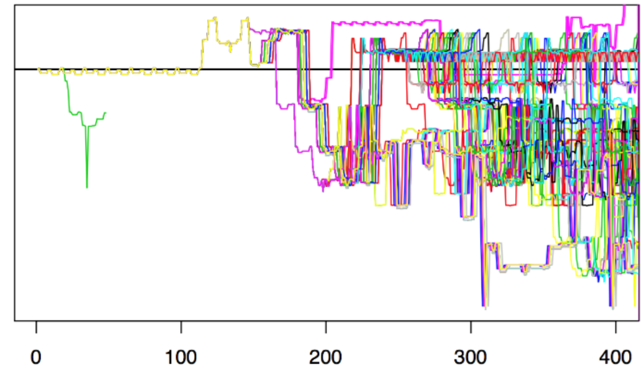


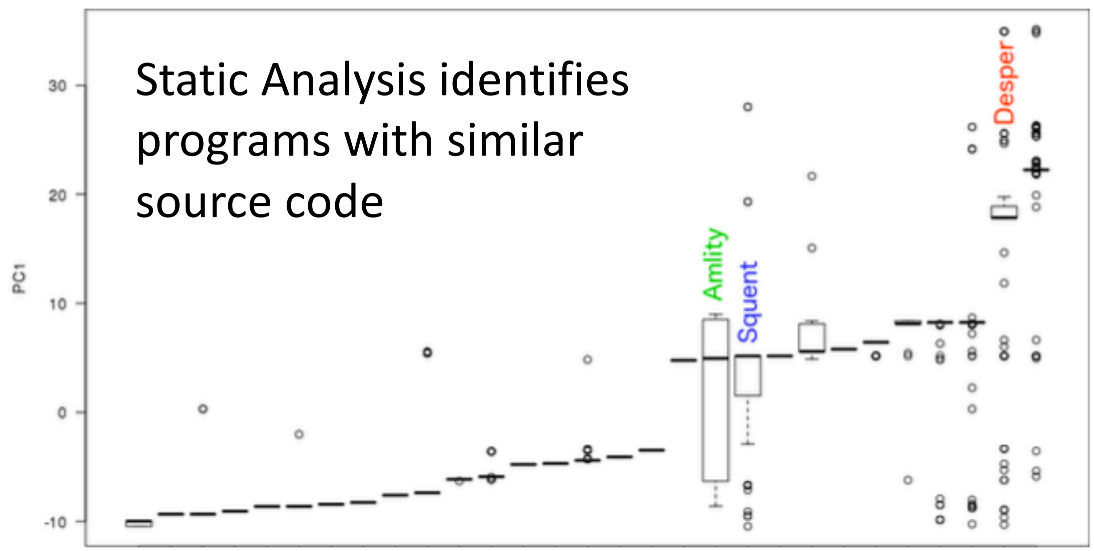
Applied ML: Malware family classification



Signal Flow graph highlights behavior relating different malware families

Program instruction analysis shows similarity and diversion of behavior

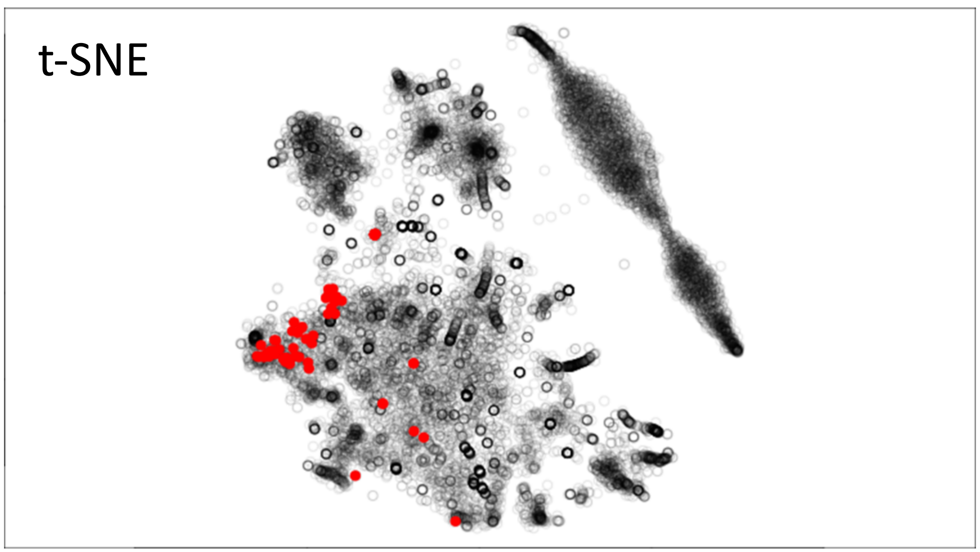


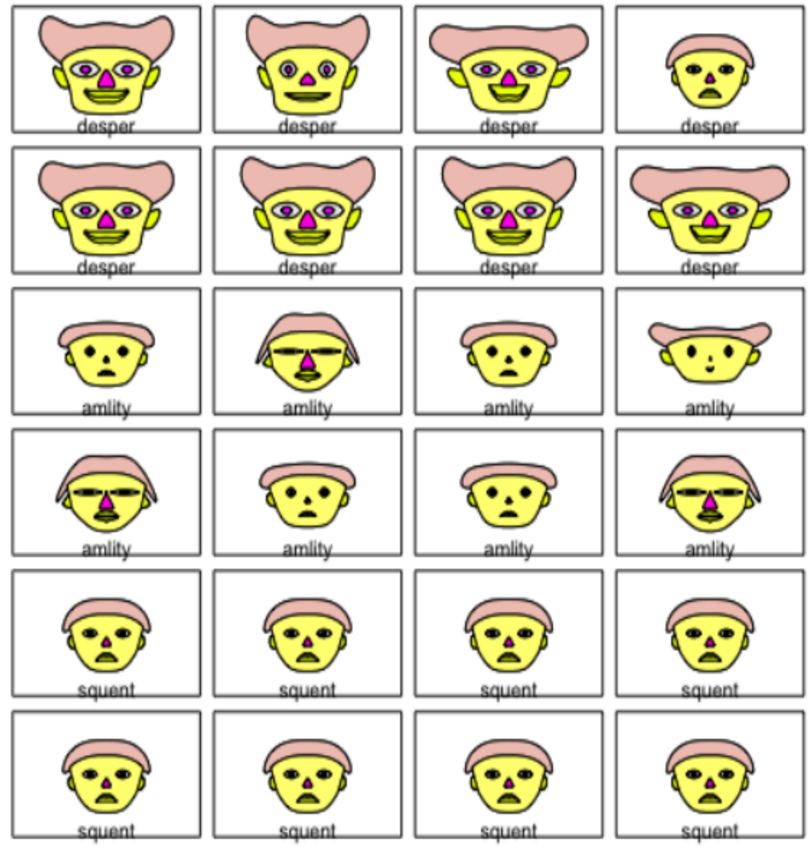


Static Analysis identifies programs with similar source code

Applied ML: Malware family classification

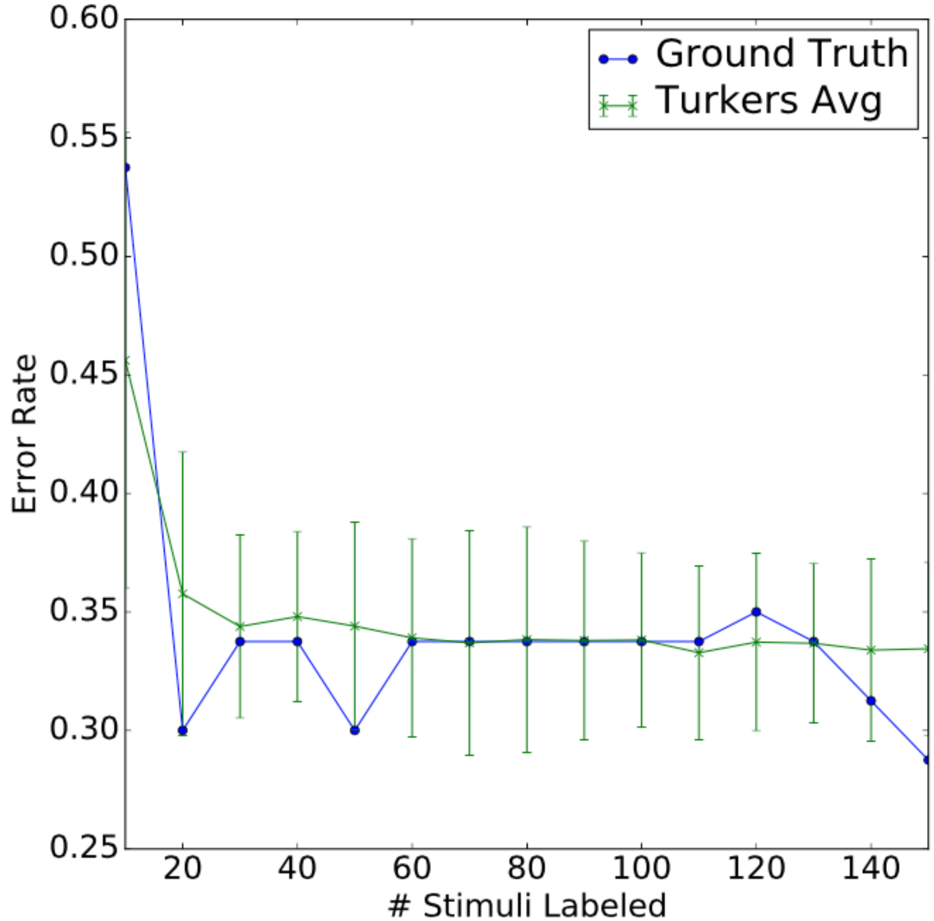
Simplify visualization of extremely complex data through the use of dimensionality reduction and associated visualization techniques





Chernoff face experiment

Applied ML: Malware family classification



- Ground Truth: SVM trained with expert ground truth labels.
- Turkers Avg: Classifier trained with layperson labels.

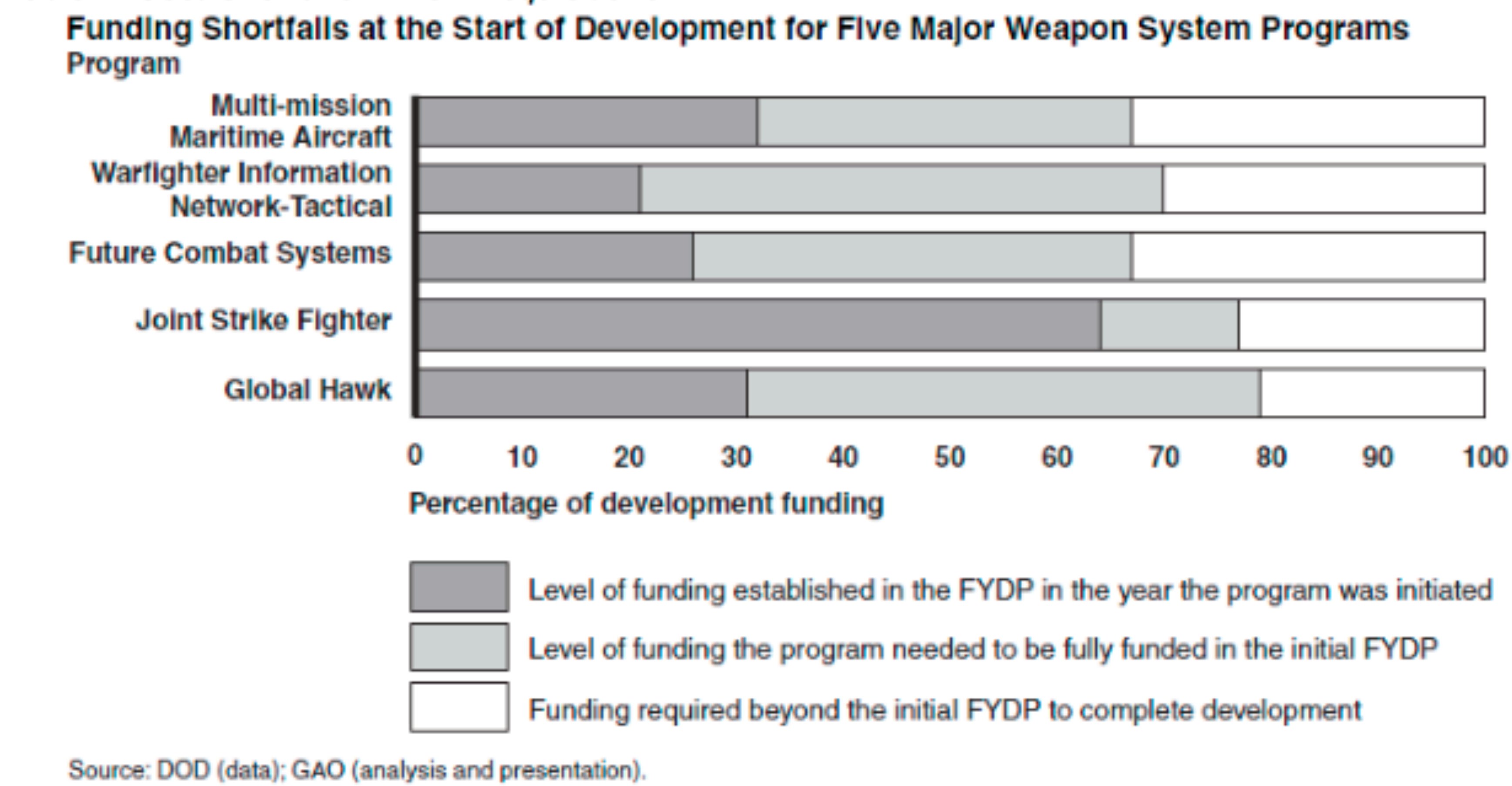
Performance surprisingly similar!

Applied ML:
Software cost estimation

Robert Ferguson, Dennis Goldenson, James McCurley, Robert W. [illegible] David Zubrow, Debra Anderson. "Quantifying Uncertainty in Early Lifecycle Cost Estimation (QUELCE)". Dec 2011. http://resou[illegible]edu/library/asset-view.cfm?assetid=10039

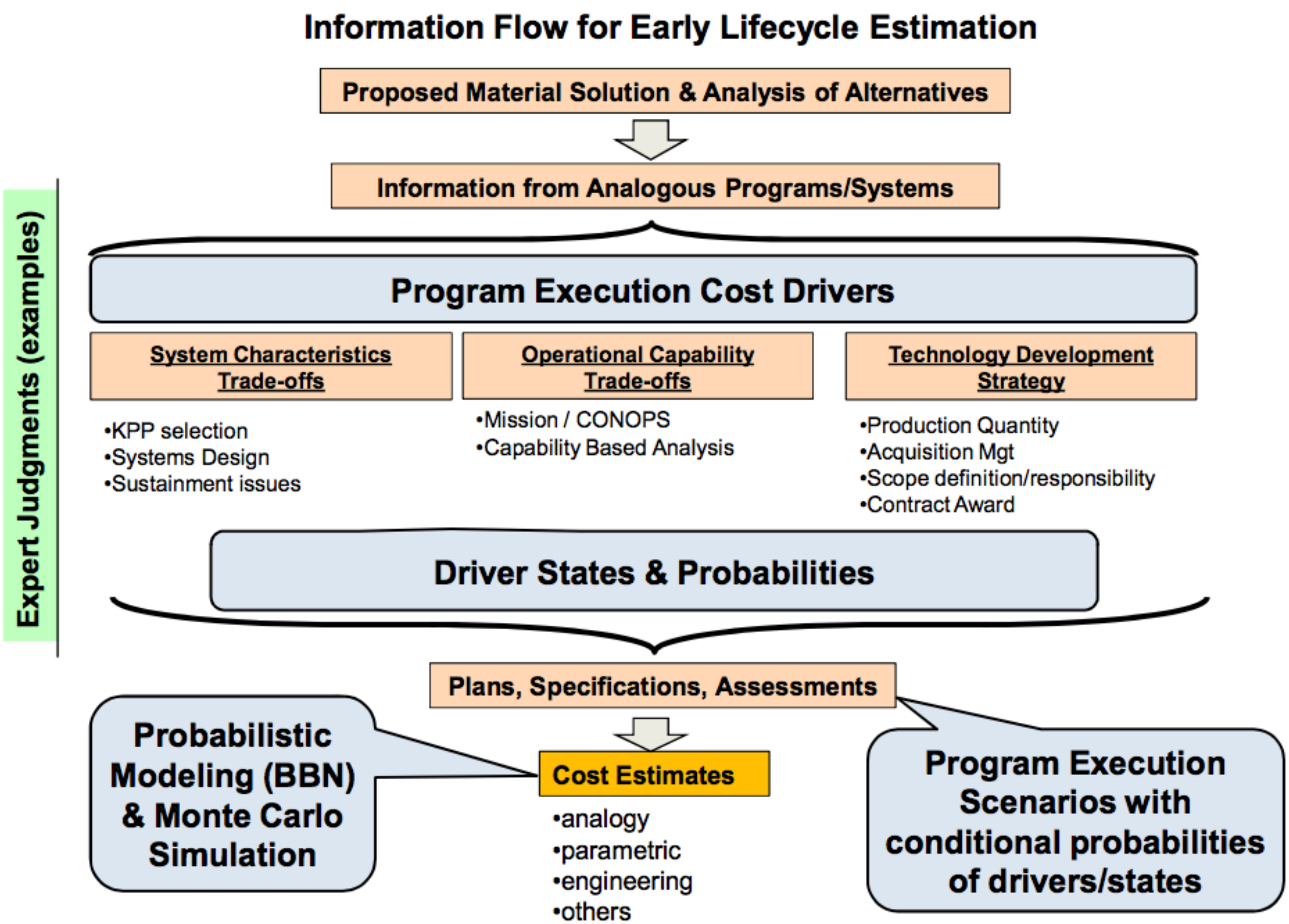
Applied ML: Software cost estimation

Table 2 Cost Overruns in DoD Acquisitions



General Accounting Office. *Defense Acquisitions: A Knowledge-Based Funding Approach Could Improve Major Weapon System Program Outcomes*. Report to the Committee on Armed Services, U.S. Senate, July 2008, GAO-08-619.

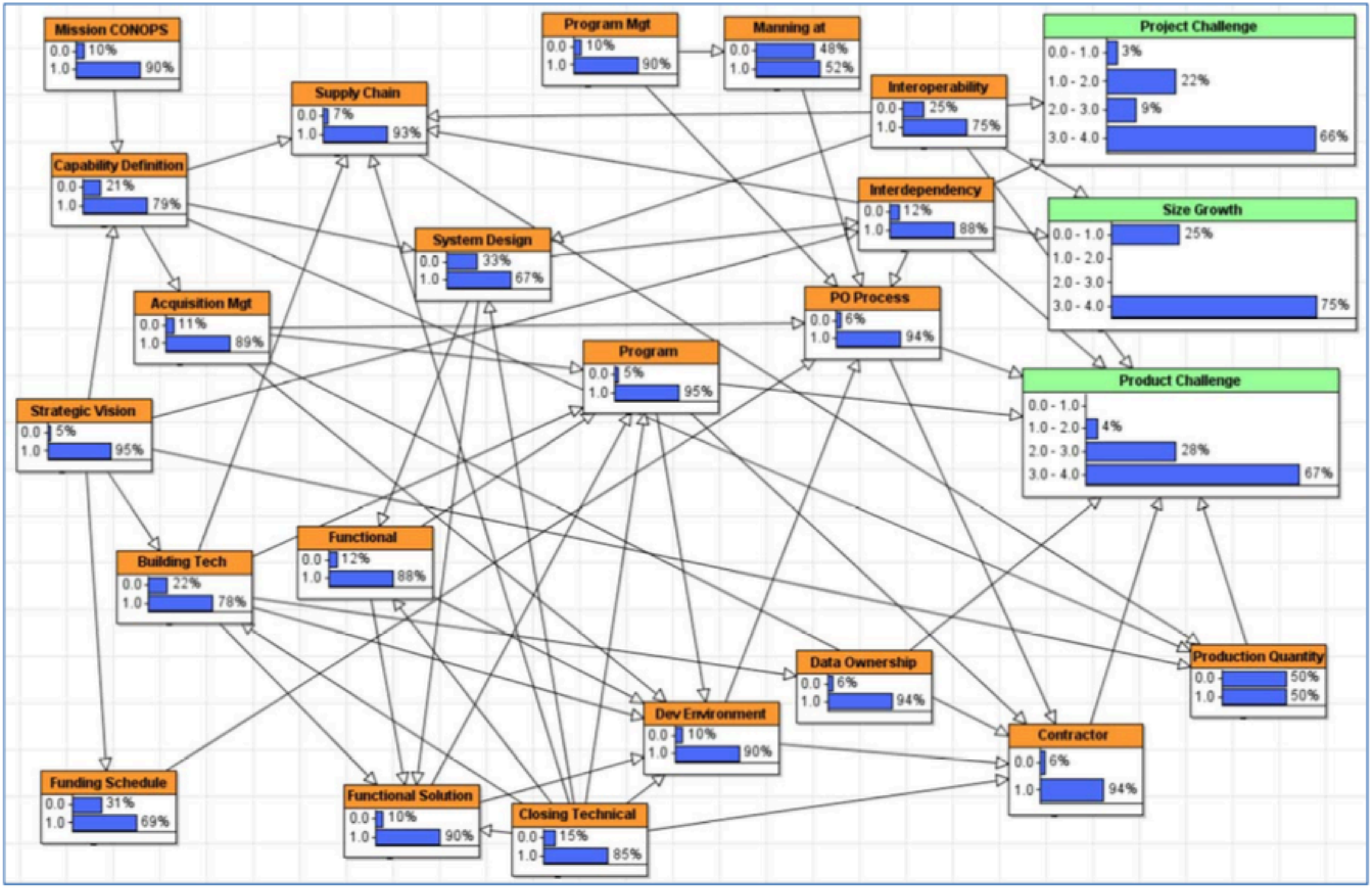
Applied ML: Software cost estimation



Applied ML: Software cost estimation

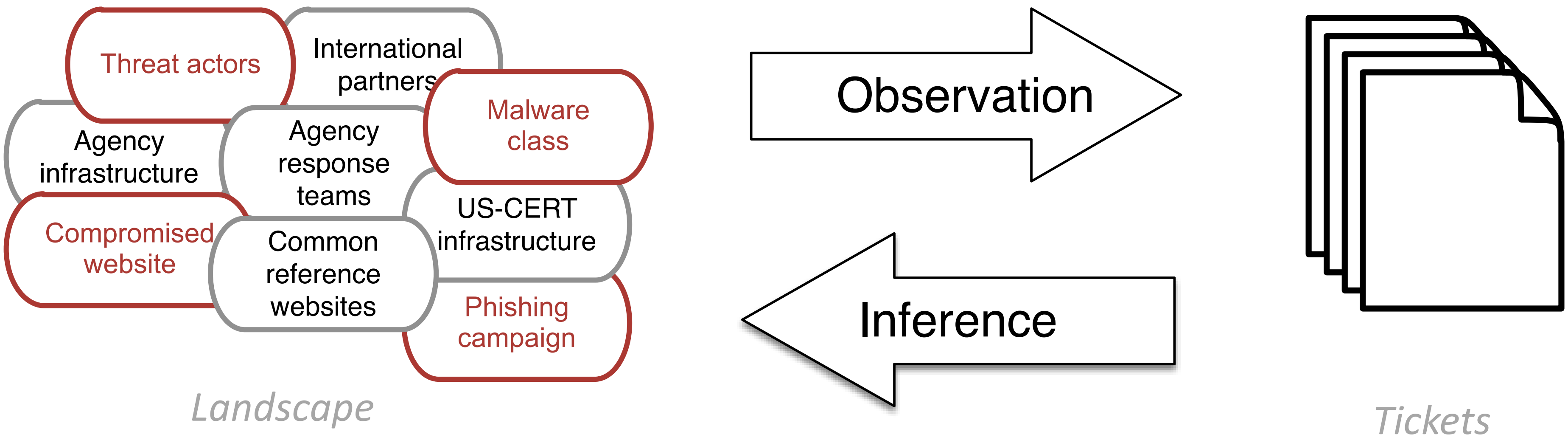
Causes \ Effects	Scope Responsibility	Scope Definition	Mission / CONOPS	Capability Definition	Funding Schedule	Prog Mgt Structure	Manning at program office	Systems Design	Standards/Certifications	Acquisition Management	Program Mgt - Contractor Relations	Project Social / Dev Env	Supply Chain Vulnerabilities	Information sharing	PO Process Performance	Sustainment Issues	Contract Award	Contractor Performance	Production Quantity	Data Ownership	Change in Strategic Vision	Advocacy Change	Industry Company Assessment	Cost Estimate	Test & Evaluation	Closing Technical Gaps (CBA)	Building Technical Capability & Capacity	Functional Measures	Functional Solution Criteria (measure)	Interdependency	Interoperability	Size	Project Challenge	Product Challenge	Total
Scope Responsibility		2				1	1			1	1	1			1																				8
Scope Definition	3					1	1																												5
Mission / CONOPS	0	3		3																															6
Capability Definition					0	0	0	3	2	2	1	1	2	2	0	1	0	0	2	0															16
Funding Schedule							1			1					2			1																	5
Prog Mgt Structure							2					1		1	2																				6
Manning at program office						2								1	2																				5
Systems Design	1												1	1		1				1			2	2	3	2	1	2	2	2	2				23
Standards/Certifications													1			1	1	1					1		3		1		1						10
Acquisition Management											2	3	1	1	2		2	2		1			1	1				1	1	1				1	20
Program Mgt - Contractor Relations												2		1	1	1		2							1		1	1	1	1	1			2	15
Project Social / Dev Env										1	1			1	2		2	2		1					1		1	1	1	1	1	1	1	1	19
Supply Chain Vulnerabilities					1			1	1	1								1	2																7
Information sharing								1							1	1		1		1						1	1								7
PO Process Performance																		2																2	4
Sustainment Issues																																			0
Contract Award																																			0
Contractor Performance																																		2	2
Production Quantity																																		2	2
Data Ownership																																		2	2
Change in Strategic Vision				3	2									2		2	2		3			3	2	3	2		3			2					29
Advocacy Change	1	2				1	1			1																									6
Industry Company Assessment																																			0
Cost Estimate																																			0
Test & Evaluation																																			0
Closing Technical Gaps (CBA)	1							3	1	1	2	2	2	1	0	2	2	2	1	1	1	2	2	2	1		2	2	2	1	1				37
Building Technical Capability & Capacity (CBA)		1				1				1	2	2	2	3		2	2	1	1	2	0	1	2	1	1	0		1	2	1					29
Functional Measures									1		2	2	1	1	1		1	1		1		1	1		2				2						17
Functional Solution Criteria (measure)									1		2	2						1					1		2					1					10
Interdependency	1	1			1	1	1	2	1		1	1	2	1	2	2		1		1	1	1	1	1	1				1		2	2	2	3	34
Interoperability	1	1						2	1	1	1	1	2	1	1	2		1		1				1	3		1		1	1		2	2	2	29
Size																																			0
Project Challenge																																			0
Product Challenge																																			0
Totals	8	10	0	6	4	7	7	12	8	10	15	18	14	17	17	15	12	19	9	10	2	8	13	11	20	3	11	8	14	11	7	5	5	17	

Applied ML: Software cost estimation

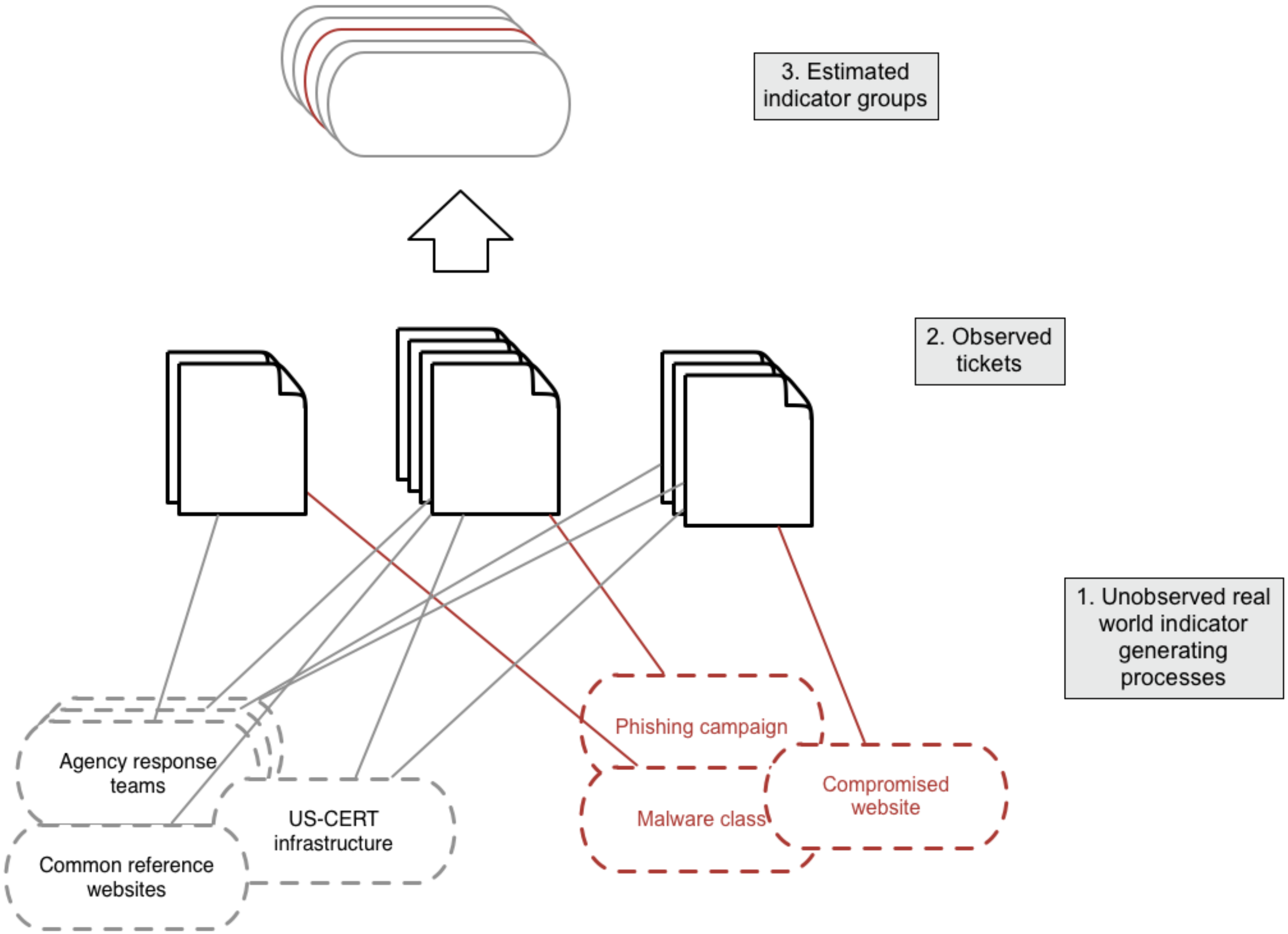


Applied ML:
Incident report mapping

Applied ML: Incident report mapping



Applied ML: Incident report mapping

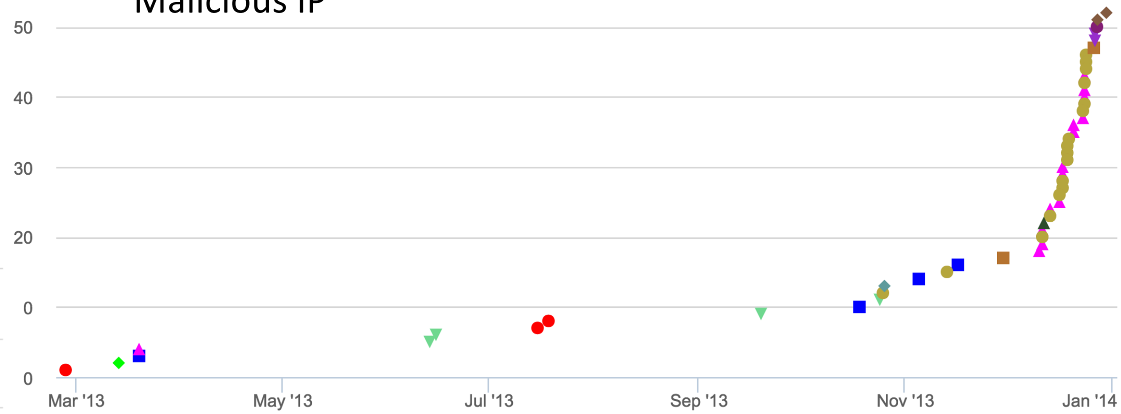


Indicators across tickets

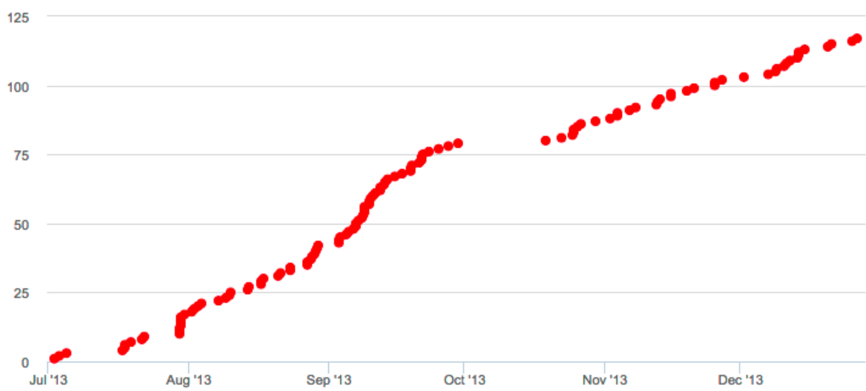
Indicators occur with diverse patterns across tickets, reporters and time.

Time on x axis, count on y axis, color coded by reporter.

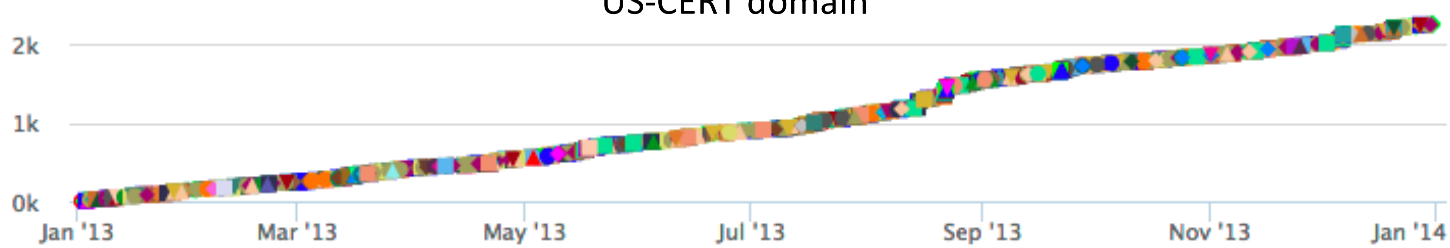
Malicious IP



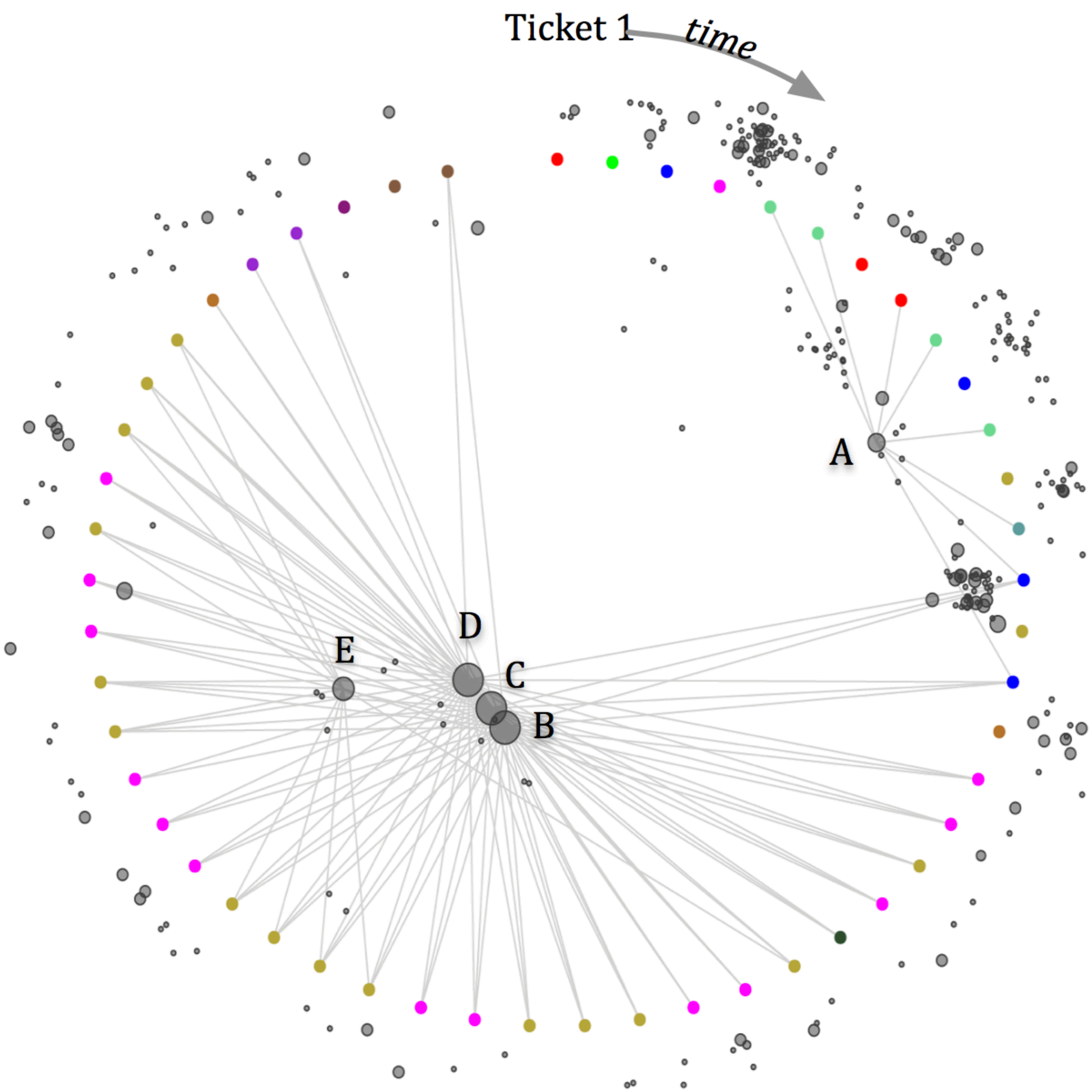
Agency IP



US-CERT domain



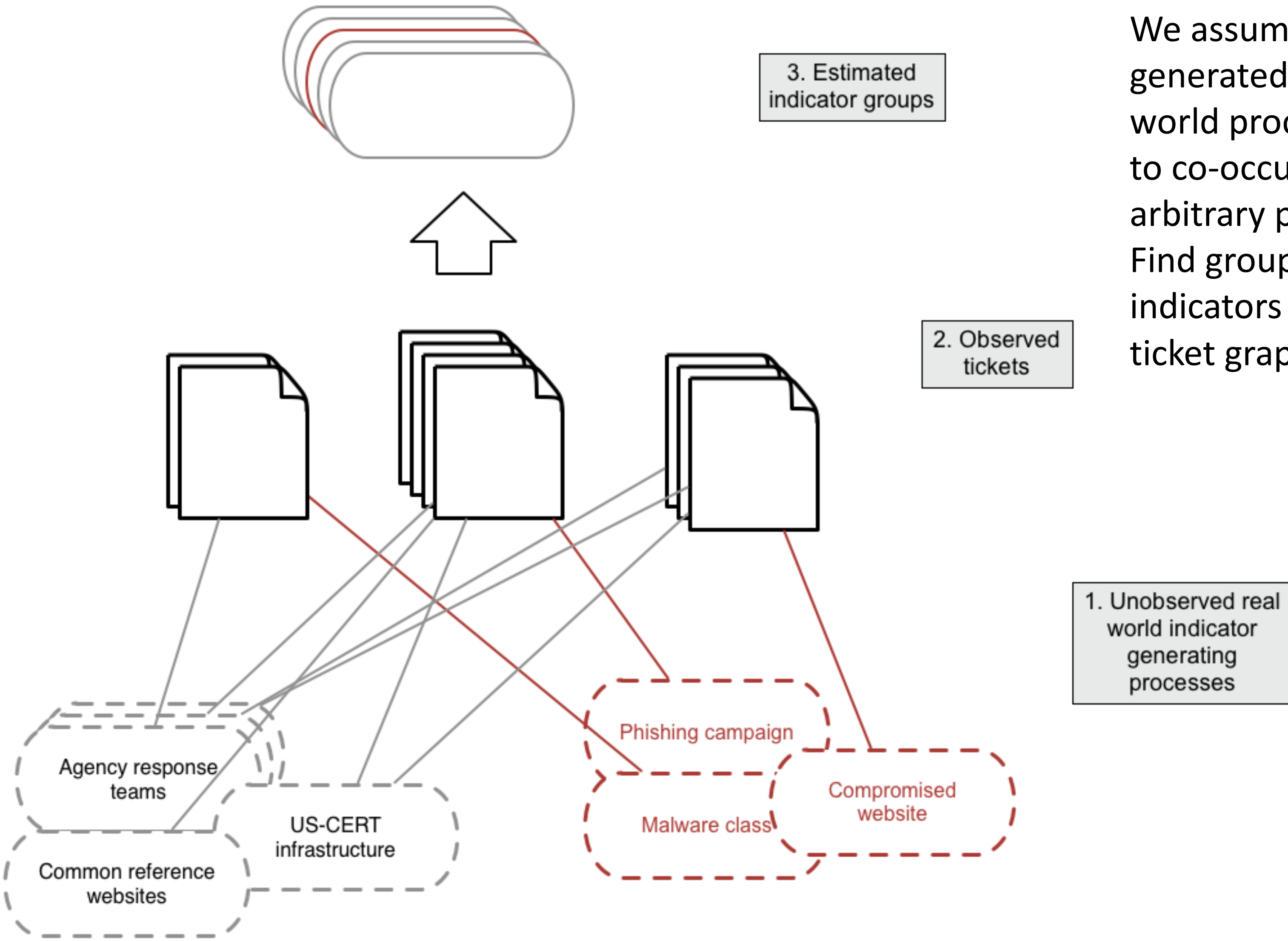
Similarity of indicators



Beginning with a reference indicator, we find indicators similar to it.
Example: a malicious IP

- Colored circles are tickets
- Grey circles are indicators
- Large indicators near center of circle have similar occurrence patterns to the reference indicator.

Indicator communities

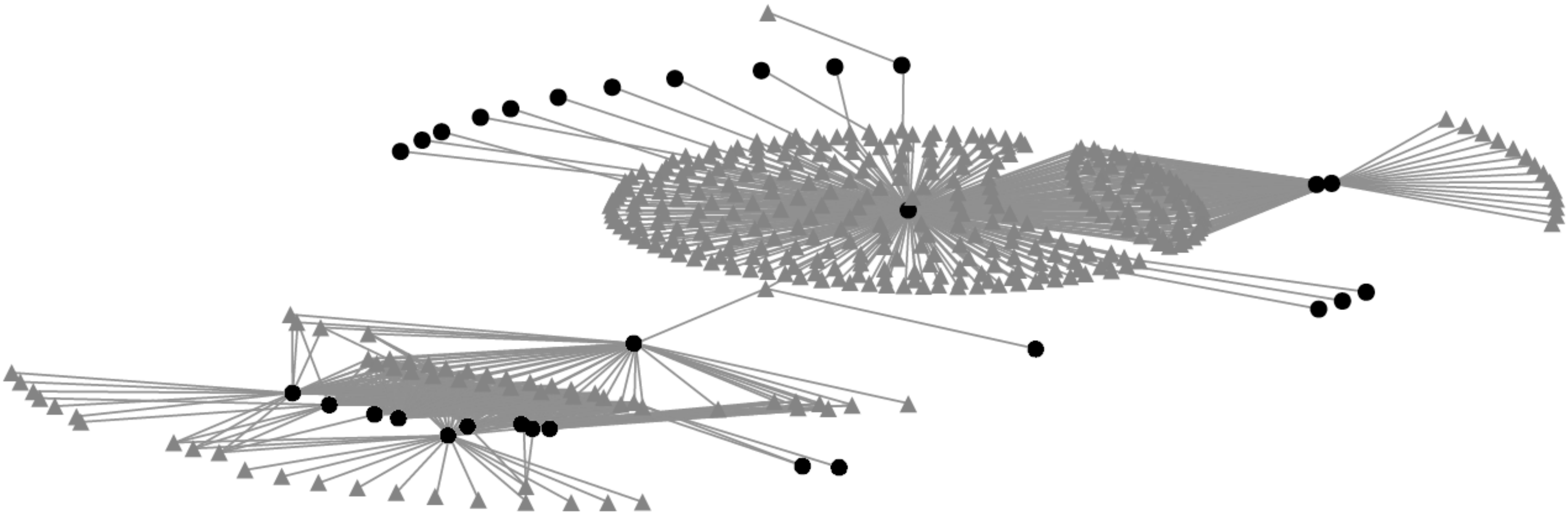


But what if we aren't starting with a reference indicator?

We assume that indicators generated by a coherent real world process will be more likely to co-occur in tickets than arbitrary pairs of indicators.

Find groups of highly similar indicators in complete indicator-ticket graph.

Indicator-ticket graph



A subset of the ticket-indicator graph
(for a small set of selected indicators)

- Tickets are grey triangles
- Indicators are black circles
- Edges connect tickets to the indicators they contain

Contact Information

Eliezer Kanal

Technical Manager

Telephone: +1 412.268.5204

Email: ekanal@sei.cmu.edu