Outside the Closed World: On Finding Intrusions with Anomaly Detection

Robin Sommer

International Computer Science Institute, &
Lawrence Berkeley National Laboratory

robin@icsi.berkeley.edu
http://www.icir.org

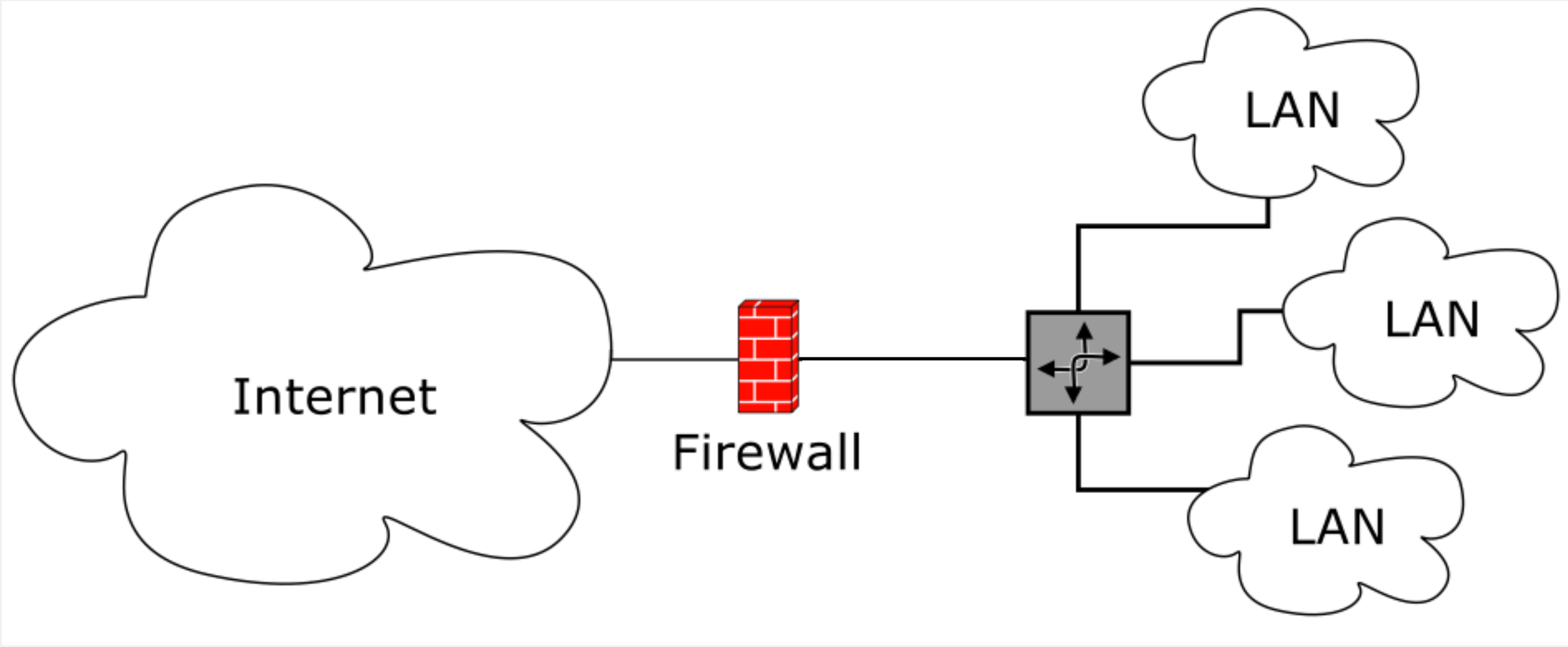
Advanced Topics in Computer Security
UC Berkeley

April 2010

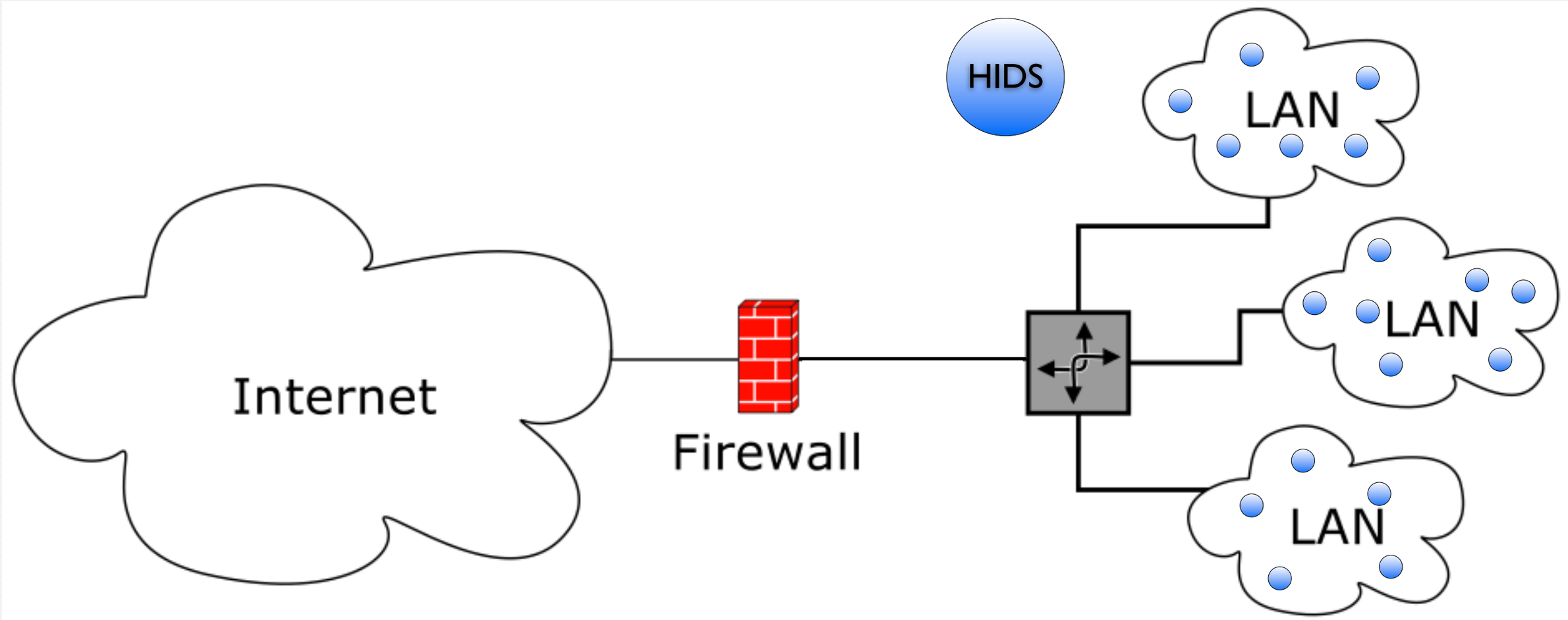
Monitoring For Intrusions

- Too many bad folks out there on the Internet.
  - Constantly scanning the Net for vulnerable systems.
  - When they mount an attack on your network, you want to know.
- Operators deploy systems that monitor their network.
  - Intrusion detection or intrusion prevention systems (IDS/IPS).
  - Terminology is a bit fuzzy these days ("security suites", "malware protection").
- How does an IDS find the attack?
  - Vantage point: host-based vs. network-based.
  - Detection approach: misuse detection vs. anomaly detection.

Achieving Visibility



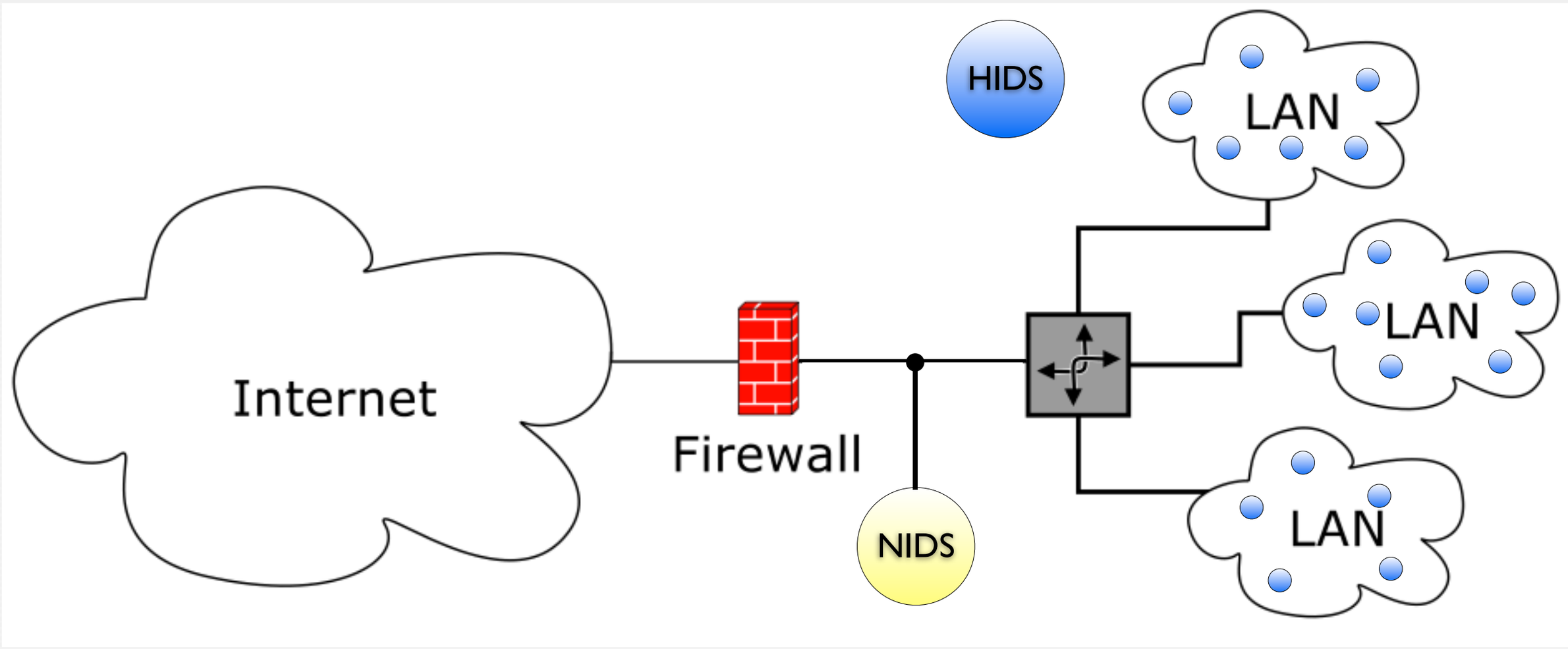
Achieving Visibility



Host-based

+ High-level semantics
+ Performance
+ Deals with crypto
− Management hassle
− Must trust host

Achieving Visibility



Host-based

+ High-level semantics
+ Performance
+ Deals with crypto
− Management hassle
− Must trust host

Network-based

+ Easy setup, with broad coverage
+ Hard to subvert
− Packets lack context
− Performance
− Does not deal with crypto

Finding Malicious Activity

Misuse detection (aka signature-/rule-based)

Searching for what we know to be bad.

Anomaly detection

Searching for what is not normal.

Specification-based detection

Searching for what we not *know* to be good.

Behavior-based detection

Searching for activity patterns based on context.

Misuse Detection With Snort

Snort is the most popular open-source NIDS.

- Available since 1999, now developed by SourceFire Inc.
- Comes with 1000s of "signatures" (although no longer open-source).

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139
    flow:to_server,established
    content:"|eb2f 5feb 4a5e 89fb 893e 89f2|"
    msg:"EXPLOIT x86 linux samba overflow"
    reference:bugtraq,1816
    reference:cve,CVE-1999-0811
    classtype:attempted-admin
```

- Conceptually simple; easy to comprehend what an alarm means.
- Signatures are updated as new attacks are discovered.
- Similar to what most commercial NIDS/NIPS do as well.

- Many attacks cannot be (realiably) defined with such a signature.
- Cannot find the "zero-days".

Misuse Detection With Bro

Bro is an open-source NIDS from Berkeley

- Developed by Vern Paxson's group at ICSI since 1996.
- Comes with a full domain-specific scripting language.
- Used most commonly for misuse-based detection (but not limited to that).

```
global ssh_hosts: set[addr];

event connection_established(c: connection)
    {
    local responder = c.id.resp_h; # Responder's address
    local service = c.id.resp_p;   # Responder's port

    if ( service != 22/tcp )
        return; # Not SSH.

    if ( responder in ssh_hosts )
        return; # We already know this one.

    add ssh_hosts[responder]; # Found a new host.
    print "New SSH host found", responder;
    }
```

Anomaly Detection

Assumption: *Attacks exhibit characteristics different from normal traffic, for a suitable definition of normal.*

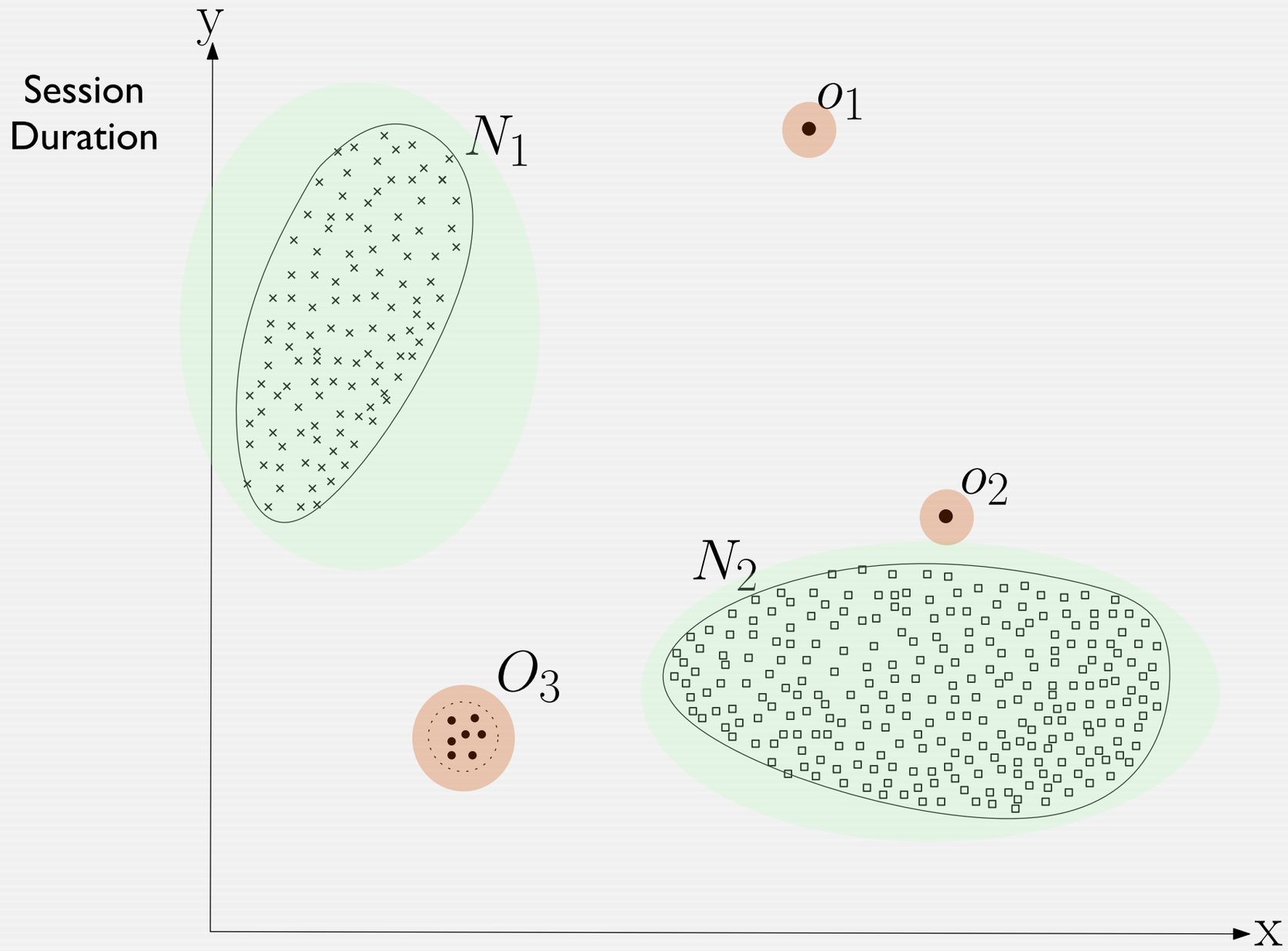
Detection has two components:

(1) Build a profile of normal activity (commonly offline).
(2) Match activity against profile and report what deviates.

Originally introduced by Denning's IDES in 1987:

- Host-level system building per-user profiles of activity.
- Login frequency, password failures, session duration, resource consumption.
- Build probability distributions for attribute/user pairs.
- Determine likelihood that new activity is outside of the assumed model.

A Simple 2D Model of Normal



Source: Chandola et al. 2009

Examples of Past Efforts

Technique Used	Section	References
Statistical Profiling using Histograms	Section 7.2.1	NIDES [Anderson et al. 1994; Anderson et al. 1995; Javitz and Valdes 1991], EMERALD [Porras and Neumann 1997], Yamanishi et al [2001; 2004], Ho et al. [1999], Kruegel at al [2002; 2003], Mahoney et al [2002; 2003; 2003; 2007], Sargor [1998] Wang and Stolfo [2004]
Parametric Statistical Modeling	Section 7.1	Gwadera et al [2005b; 2004], Ye and Chen [2001]
Non-parametric Statistical Modeling	Section 7.2.2	Chow and Yeung [2002]
Bayesian Networks	Section 4.2	Siaterlis and Maglaris [2004], Sebyala et al. [2002], Valdes and Skinner [2000], Bronstein et al. [2001]
Neural Networks	Section 4.1	HIDE [Zhang et al. 2001], NSOM [Labib and Vemuri 2002], Smith et al. [2002], Hawkins et al. [2002], Kruegel et al. [2003], Manikopoulos and Papavassiliou [2002], Ramadas et al. [2003]
Support Vector Machines	Section 4.3	Eskin et al. [2002]
Rule-based Systems	Section 4.4	ADAM [Barbara et al. 2001a; Barbara et al. 2003; Barbara et al. 2001b], Fan et al. [2001], Helmer et al. [1998], Qin and Hwang [2004], Salvador and Chan [2003], Otey et al. [2003]
Clustering Based	Section 6	ADMIT [Sequeira and Zaki 2002], Eskin et al. [2002], Wu and Zhang [2003], Otey et al. [2003]
Nearest Neighbor based	Section 5	MINDS [Ertoz et al. 2004; Chandola et al. 2006], Eskin et al. [2002]
Spectral	Section 9	Shyu et al. [2003], Lakhina et al. [2005], Thottan and Ji [2003],Sun et al. [2007]
Information Theoretic	Section 8	Lee and Xiang [2001],Noble and Cook [2003]

Examples of techniques used for network intrusion detection.

Source: Chandola et al. 2009

Examples of Past Efforts

Technique Used	Section	References
Statistical Profiling using Histograms	Section 7.2.1	NIDES [Anderson et al. 1994; Anderson et al. 1995; Javitz and Valdes 1991], EMERALD [Porras and Neumann 1997], Yamanishi et al [2001; 2004], Ho et al. [1999], Kruegel at al [2002; 2003], Mahoney et al [2002; 2003; 2003; 2007], Sargor [1998] Wang and Stolfo [2004]
Parametric Statistical Modeling	Section 7.1	Gwadera et al [2005b; 2004], Ye and Chen [2001]
Non-parametric Statistical Modeling		Barbara et al. 2001b], Fan et al. [2001], Helmer et al. [1998], Qin and Hwang [2004], Salvador and Chan [2003], Otey et al. [2003]
Bayesian Networks		
Neural Networks		
Support Vector Machines		
Rule-based Systems		
Clustering Based	Section 6	ADMIT [Sequeira and Zaki 2002], Eskin et al. [2002], Wu and Zhang [2003], Otey et al. [2003]
Nearest Neighbor based	Section 5	MINDS [Ertoz et al. 2004; Chandola et al. 2006], Eskin et al. [2002]
Spectral	Section 9	Shyu et al. [2003], Lakhina et al. [2005], Thottan and Ji [2003],Sun et al. [2007]
Information Theoretic	Section 8	Lee and Xiang [2001],Noble and Cook [2003]

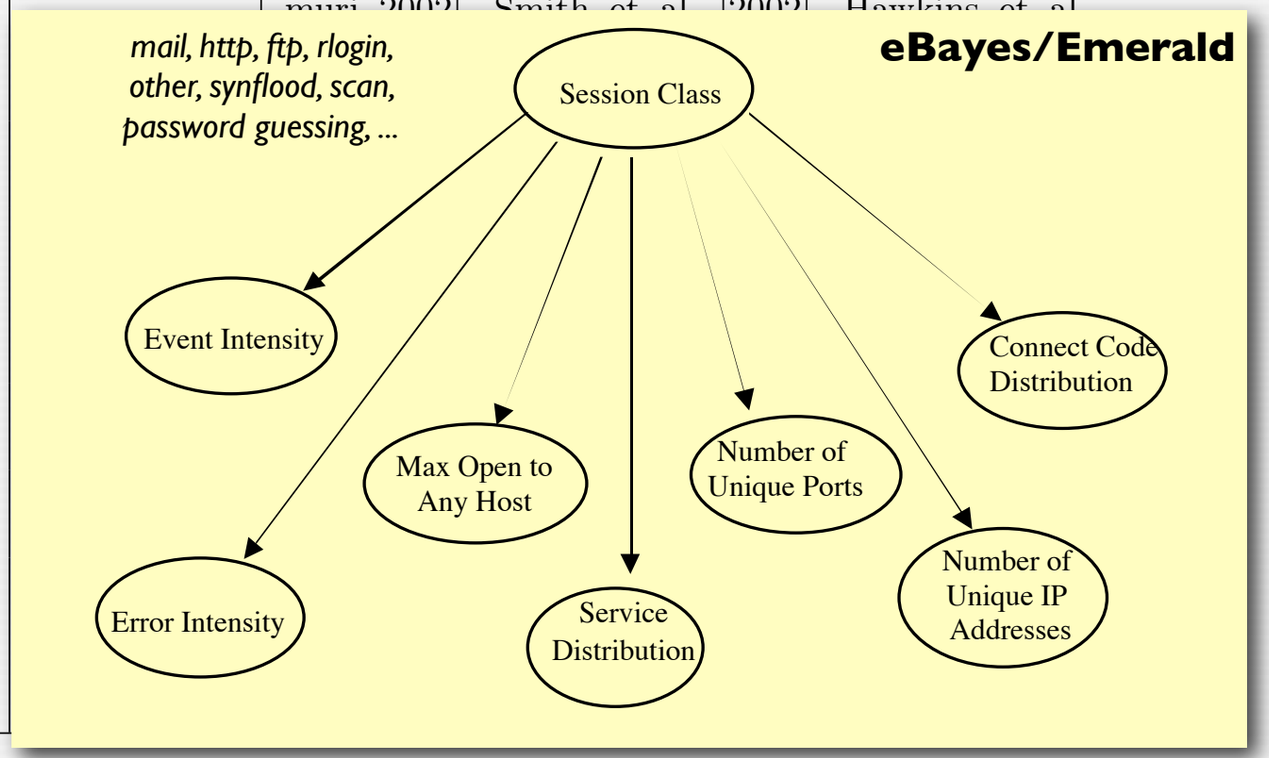
Feature name	Feature description **MINDS**
count-dest	Number of flows to unique destination IP addresses inside the network in the last T seconds from the same source
count-src	Number of flows from unique source IP addresses inside the network in the last T seconds to the same destination
count-serv-src	Number of flows from the source IP to the same destination port in the last T seconds
count-serv-dest	Number of flows to the destination IP address using same source port in the last T seconds

Examples of techniques used for network intrusion detection.

Source: Chandola et al. 2009

Examples of Past Efforts

Technique Used	Section	References
Statistical Profiling using Histograms	Section 7.2.1	NIDES [Anderson et al. 1994; Anderson et al. 1995; Javitz and Valdes 1991], EMERALD [Porras and Neumann 1997], Yamanishi et al [2001; 2004], Ho et al. [1999], Kruegel at al [2002; 2003], Mahoney et al [2002; 2003; 2003; 2007], Sargor [1998] Wang and Stolfo [2004]
Parametric Statistical Modeling	Section 7.1	Gwadera et al [2005b; 2004], Ye and Chen [2001]
Non-parametric Statistical Modeling	Section 7.2.2	Chow and Yeung [2002]
Bayesian Networks	Section 4.2	Siaterlis and Maglaris [2004], Sebyala et al. [2002], Valdes and Skinner [2000], Bronstein et al. [2001]
Neural Networks	Section 4.1	HIDE [Zhang et al. 2001], NSOM [Labib and Vemuri 2002], Smith et al. [2002], Hawkins et al
Support Vector Machines		
Rule-based Systems		
Clustering Based		
Nearest Neighbor based		
Spectral		
Information Theoretic		

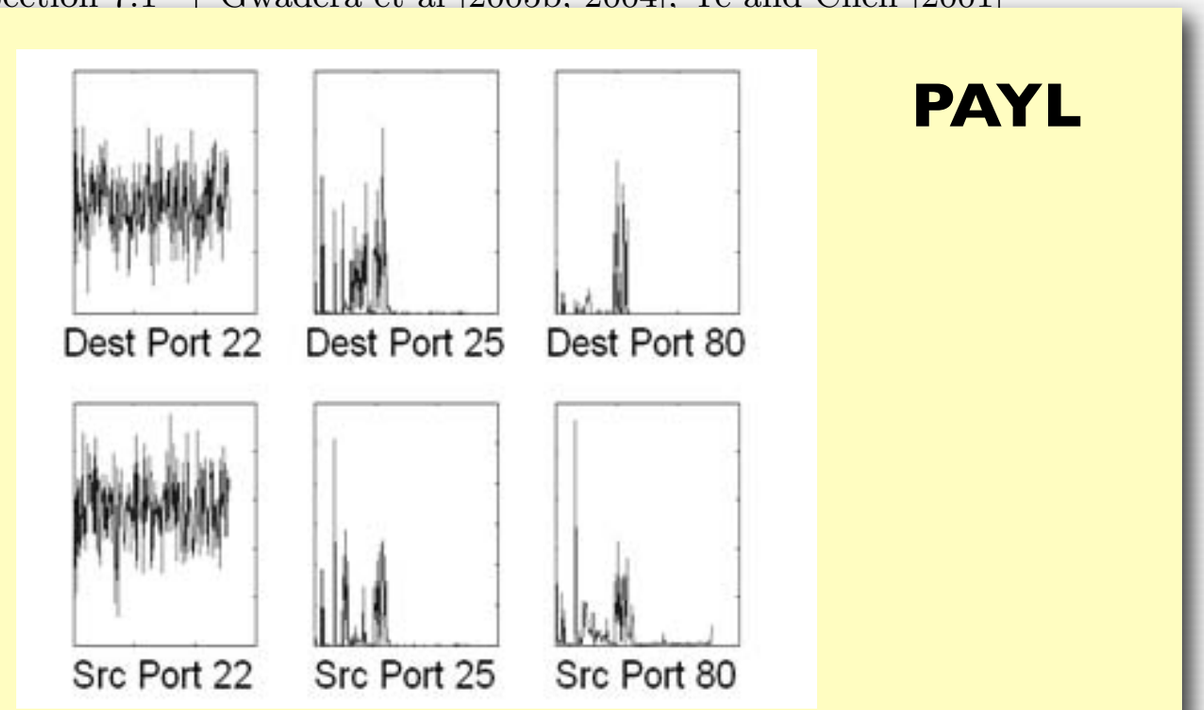


Examples of techniques used for network intrusion detection.

Source: Chandola et al. 2009

Examples of Past Efforts

Technique Used	Section	References
Statistical Profiling using Histograms	Section 7.2.1	NIDES [Anderson et al. 1994; Anderson et al. 1995; Javitz and Valdes 1991], EMERALD [Porras and Neumann 1997], Yamanishi et al [2001; 2004], Ho et al. [1999], Kruegel at al [2002; 2003], Mahoney et al [2002; 2003; 2003; 2007], Sargor [1998] Wang and Stolfo [2004]
Parametric Statistical Modeling	Section 7.1	Gwadera et al [2005b; 2004], Ye and Chen [2001]
Non-parametric Statistical Modeling		
Bayesian Networks		
Neural Networks		
Support Vector Machines		
Rule-based Systems		
Clustering Based		
Nearest Neighbor based	Section 5	MINDS [Ertoz et al. 2004; Chandola et al. 2006], Eskin et al. [2002]
Spectral	Section 9	Shyu et al. [2003], Lakhina et al. [2005], Thottan and Ji [2003],Sun et al. [2007]
Information Theoretic	Section 8	Lee and Xiang [2001],Noble and Cook [2003]



Examples of techniques used for network intrusion detection.

Source: Chandola et al. 2009

Examples of Past Efforts

Technique Used	Section	References
Statistical Profiling using Histograms	Section 7.2.1	NIDES [Anderson et al. 1994; Anderson et al. 1995; Javitz and Valdes 1991], EMERALD [Porras and Neumann 1997], Yamanishi et al [2001; 2004], Ho et al. [1999], Kruegel at al [2002; 2003], Mahoney et al [2002; 2003; 2003; 2007], Sargor [1998] Wang and
Parametric Statistical Modeling	Section 7.1	Gwadera et al [2005b; 2004], Ye and Chen [2001]
Non-parametric Statistical Modeling	Section 7.2.2	Chow and Yeung [2002]
Bayesian Networks	Section 4.2	Siaterlis and Maglaris [2004], Sebyala et al. [2002], Valdes and Skinner [2000], Bronstein et al. [2001]
Neural Networks	Section 4.1	HIDE [Zhang et al. 2001], NSOM [Labib and Vemuri 2002], Smith et al. [2002], Hawkins et al. [2002], Kruegel et al. [2003], Manikopoulos and Papavassiliou [2002], Ramadas et al. [2003]
Support Vector Machines	Section 4.3	Eskin et al. [2002]
Rule-based Systems	Section 4.4	ADAM [Barbara et al. 2001a; Barbara et al. 2003; Barbara et al. 2001b], Fan et al. [2001], Helmer et al. [1998], Qin and Hwang [2004], Salvador and Chan [2003], Otey et al. [2003]
Clustering Based	Section 6	ADMIT [Sequeira and Zaki 2002], Eskin et al. [2002], Wu and Zhang [2003], Otey et al. [2003]
Nearest Neighbor based	Section 5	MINDS [Ertoz et al. 2004; Chandola et al. 2006], Eskin et al. [2002]
Spectral	Section 9	Shyu et al. [2003], Lakhina et al. [2005], Thottan and Ji [2003],Sun et al. [2007]
Information Theoretic	Section 8	Lee and Xiang [2001],Noble and Cook [2003]

Features used

- packet sizes
- IP addresses
- ports
- header fields
- timestamps
- inter-arrival times
- session size
- session duration
- session volume
- payload frequencies
- payload tokens
- payload pattern
- ...

Examples of techniques used for network intrusion detection.

Source: Chandola et al. 2009

Examples of Past Efforts

Technique Used	Section	References
Statistical Profiling using Histograms	Section 7.2.1	NIDES [Anderson et al. 1994; Anderson et al. 1995; Javitz and Valdes 1991], EMERALD [Porras and Neumann 1997], Yamanishi et al [2001; 2004], Ho et al. [1999], Kruegel at al [2002; 2003], Mahoney et al [2002; 2003; 2003; 2007], Sargor [1998] Wang and Stolfo [2004]
Parametric Statistical Modeling	Section 7.1	Gwadera et al [2005b; 2004], Ye and Chen [2001]
Non-parametric Statistical Modeling	Section 7.2.2	Chow and Yeung [2002]
		pavassiliou [2002], Ramadas et al. [2003]
Support Vector Machines	Section 4.3	Eskin et al. [2002]
Rule-based Systems	Section 4.4	ADAM [Barbara et al. 2001a; Barbara et al. 2003; Barbara et al. 2001b], Fan et al. [2001], Helmer et al. [1998], Qin and Hwang [2004], Salvador and Chan [2003], Otey et al. [2003]
Clustering Based	Section 6	ADMIT [Sequeira and Zaki 2002], Eskin et al. [2002], Wu and Zhang [2003], Otey et al. [2003]
Nearest Neighbor based	Section 5	MINDS [Ertoz et al. 2004; Chandola et al. 2006], Eskin et al. [2002]
Spectral	Section 9	Shyu et al. [2003], Lakhina et al. [2005], Thottan and Ji [2003],Sun et al. [2007]
Information Theoretic	Section 8	Lee and Xiang [2001],Noble and Cook [2003]

```
GET /scripts/access.pl?user=johndoe&cred=admin
GET /scripts/access.pl?user=johndoe;SELECT+passwd+FROM+credentials&...
```

Web-based Attacks

Examples of techniques used for network intrusion detection.

Source: Chandola et al. 2009

The Holy Grail ...

- Anomaly detection is extremely appealing.
  - We find novel attacks without anticipating any specifics ("zero-day").
  - It's plausible: machine-learning works so well in many other domains.
- Many research efforts have explored the notion.
  - Numerous papers have been written ...
- But guess what's used *in operation*? Snort.
  - We find hardly any machine-learning-based NIDS in real-world deployments.
- Could anomaly detection be harder than it appears?

Prerequisites

- My definition of "anomaly detection" is intrusion detection based on a machine-learning algorithm.
  - Technically, the terminology is more fuzzy but that's what people associate.
- I'll focus on network-based approaches.
  - But much of the discussion applies to host-based systems as well.
- I won't tell you how machine-learning works.
  - Rather why it's difficult for this domain.
- Intrusion-detection is all about the real-world.
  - Nothing is perfect; all these systems are based on a set of heuristics.
  - Whatever helps the operator is good.

Why Is Anomaly Detection Hard?

Intrusion Detection Is Different

The intrusion detection domain faces challenges that make it fundamentally different from other fields.

Intrusion Detection Is Different

The intrusion detection domain faces challenges that make it fundamentally different from other fields.

Outlier detection and the high costs of errors

How do we find the opposite of normal?

Interpretation of results

What does that anomaly *mean*?

Evaluation

How do we make sure it actually works?

Training data

What do we train our system with?

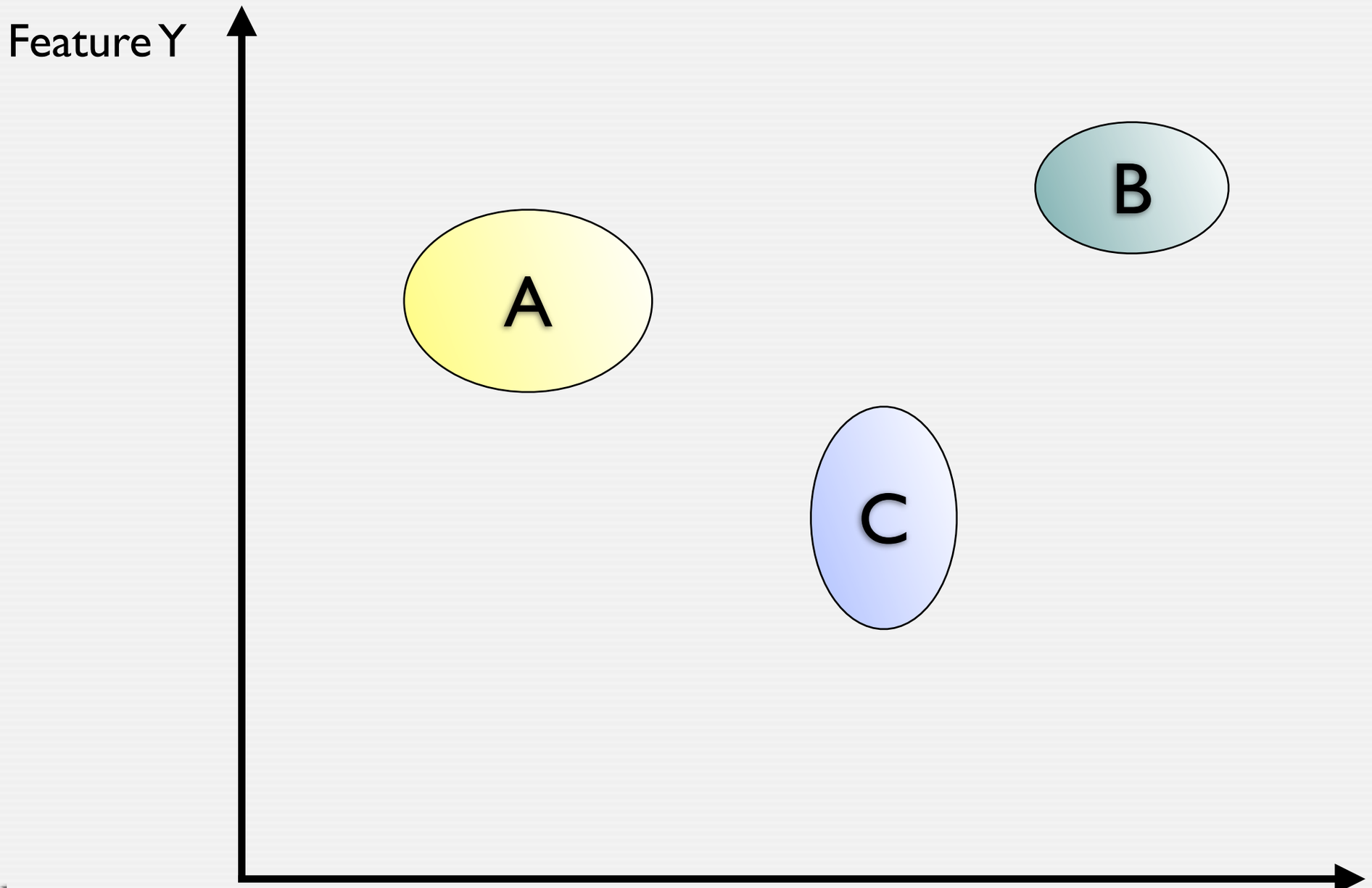
Evasion risk

Can the attacker mislead our system?

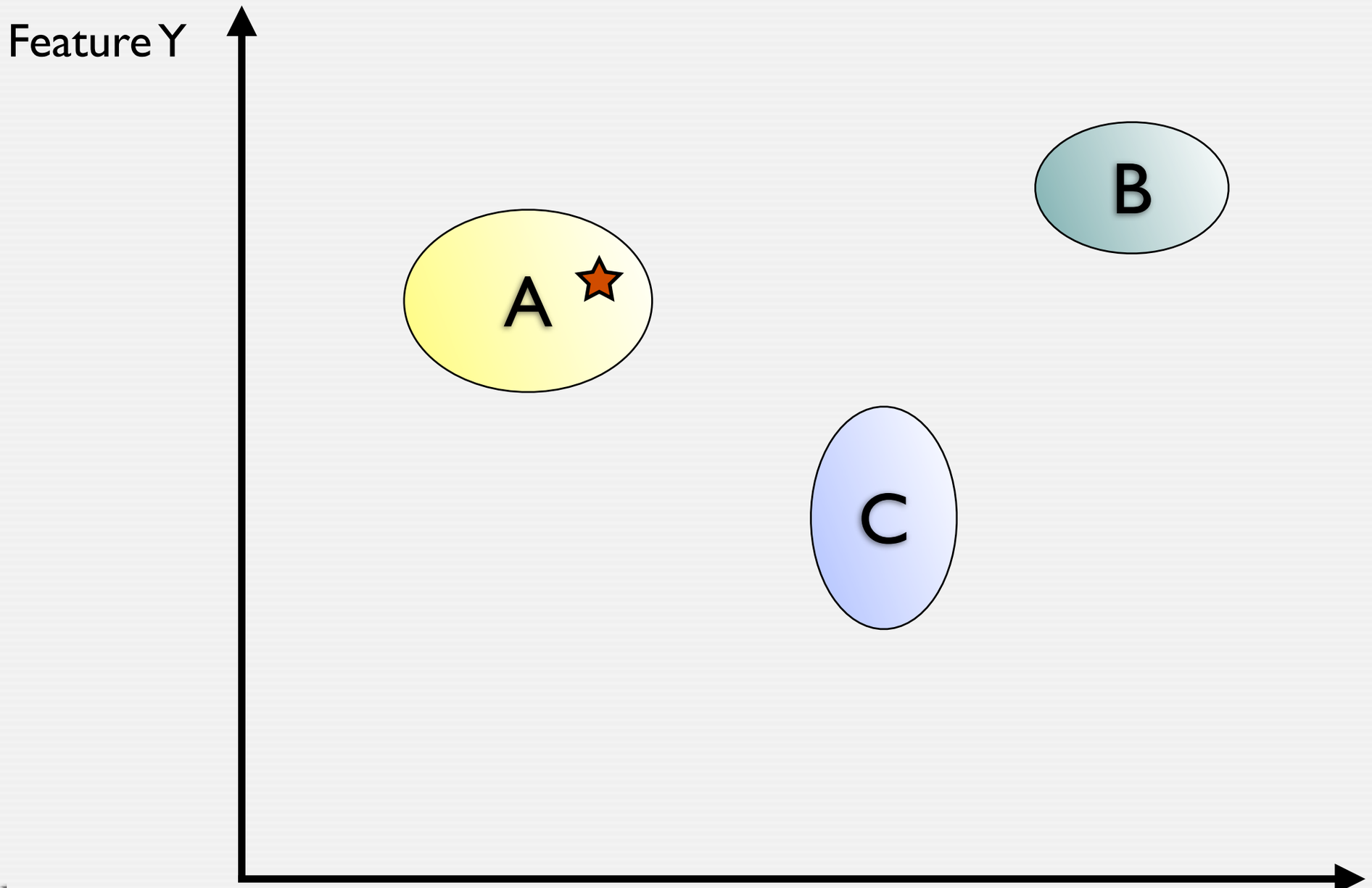
Outlier Detection

- Anomaly detection is *outlier detection*.
  - Machine-learning builds a model of its normal training data.
  - Given an observation, decide whether it fits the model.
- Problem: Machine-learning is not that good at this.
  - It's better at finding similarity than abnormality.
  - The classical machine-learning application is *classification*.
  - Training is done with equally *many* specimen of all categories

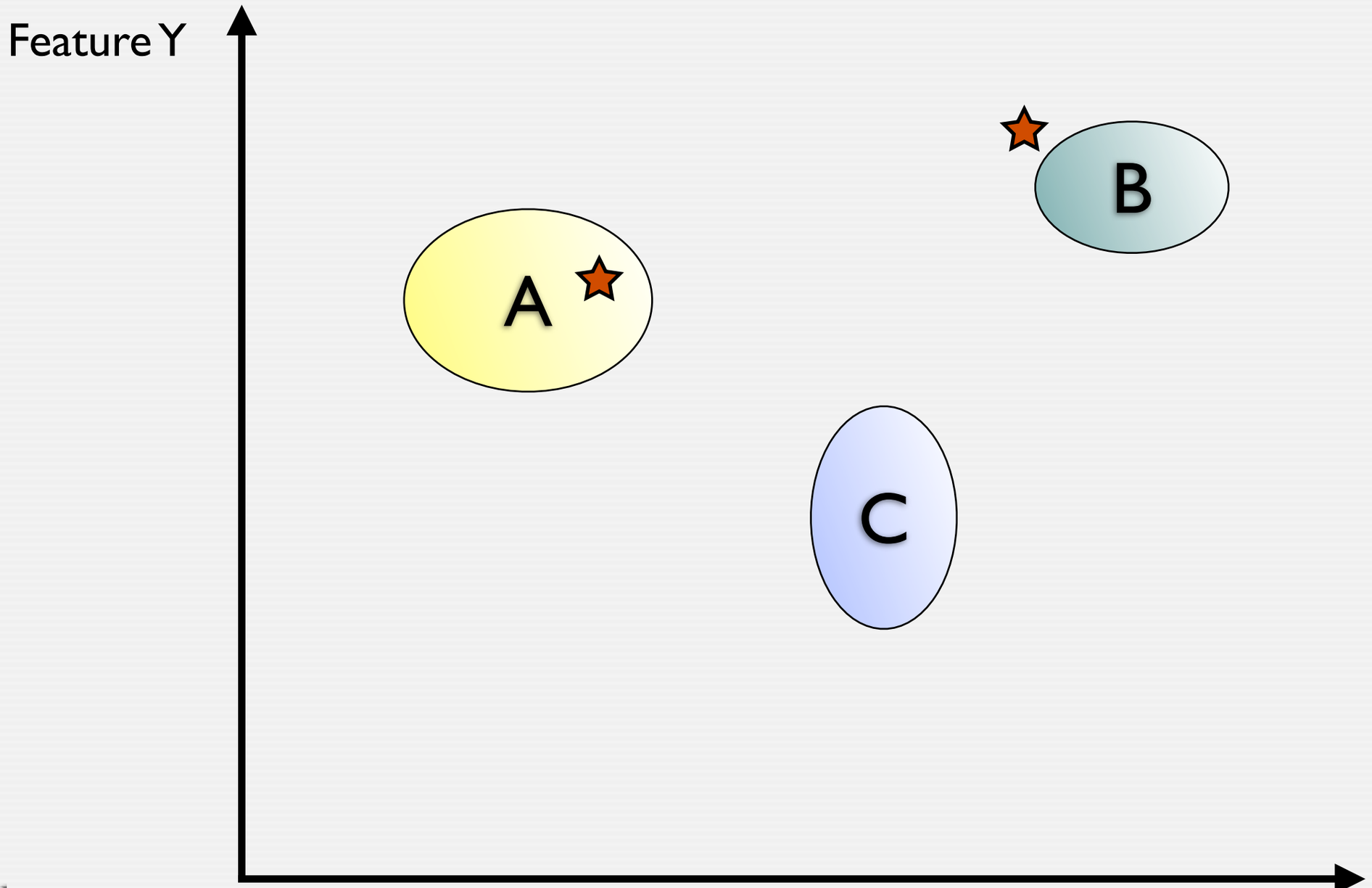
Classification Problem



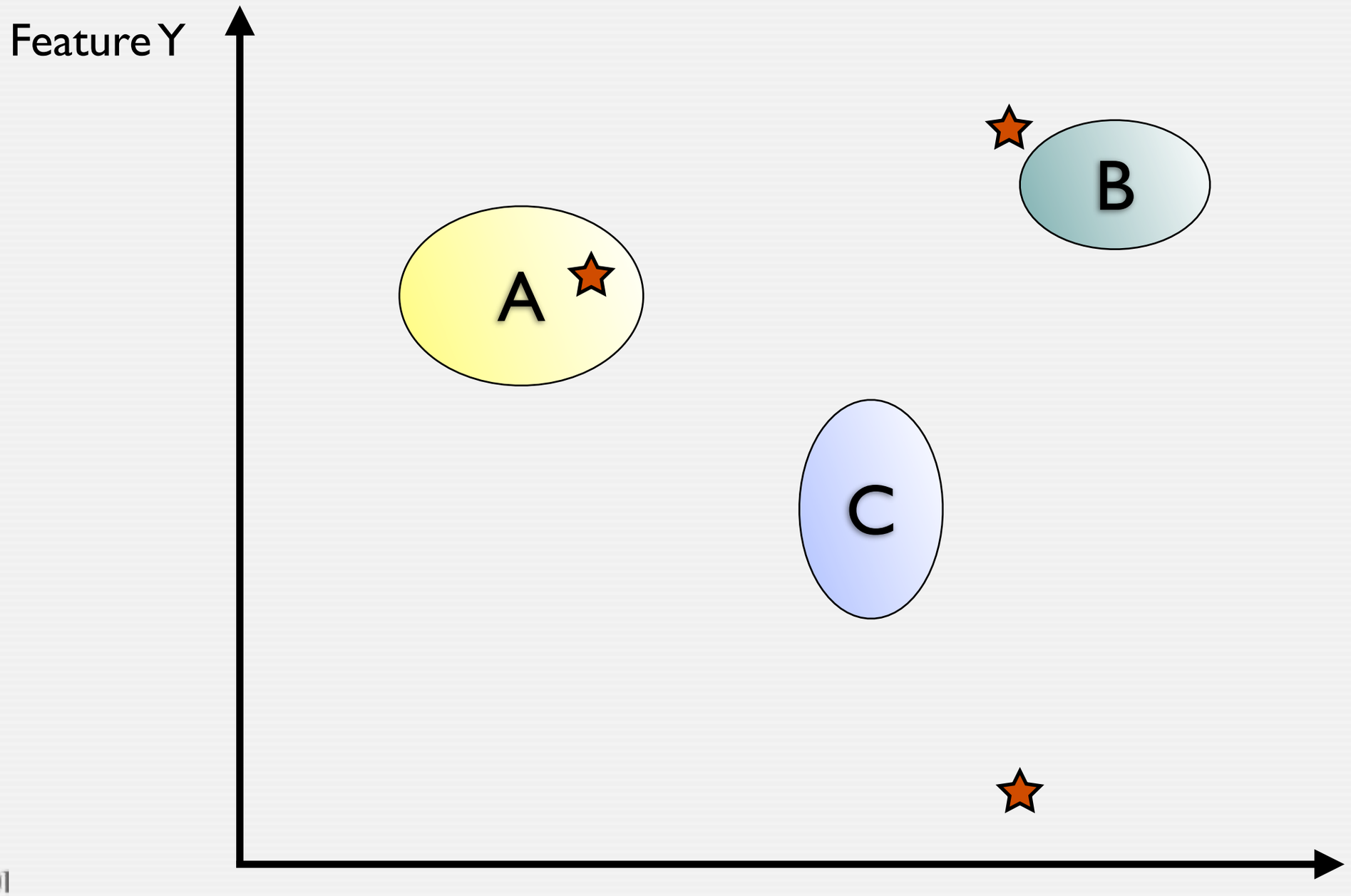
Classification Problem



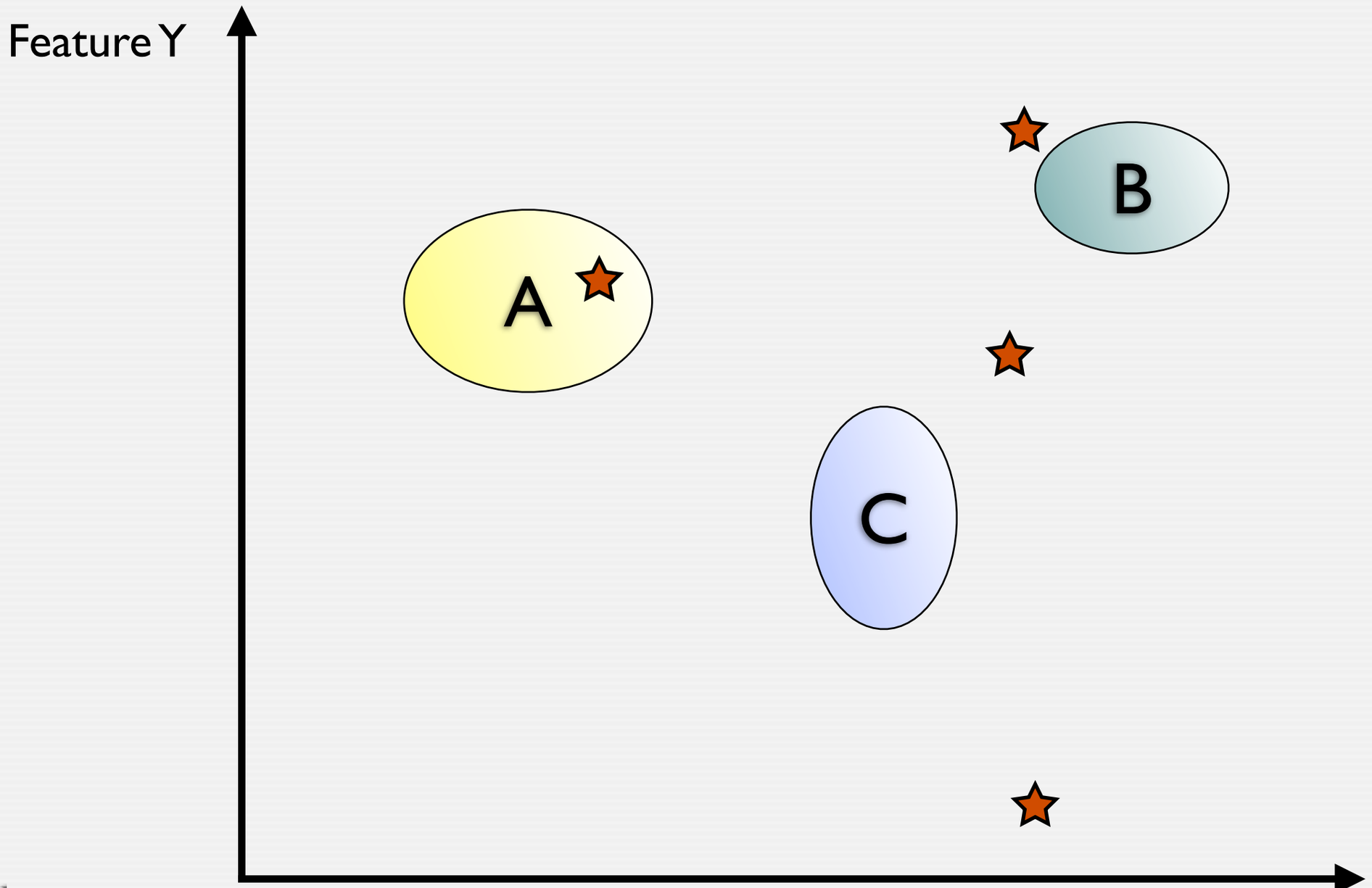
Classification Problem



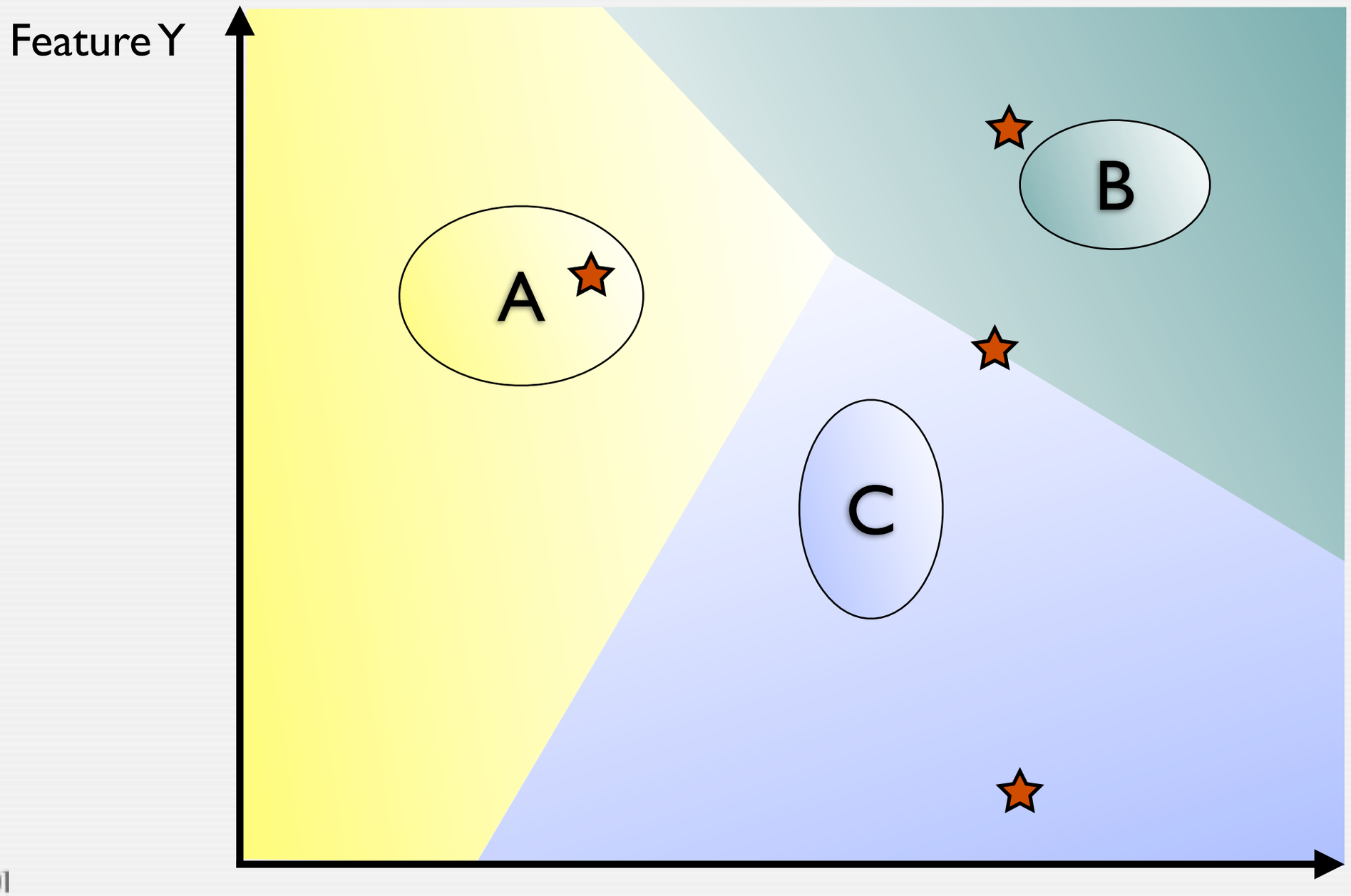
Classification Problem



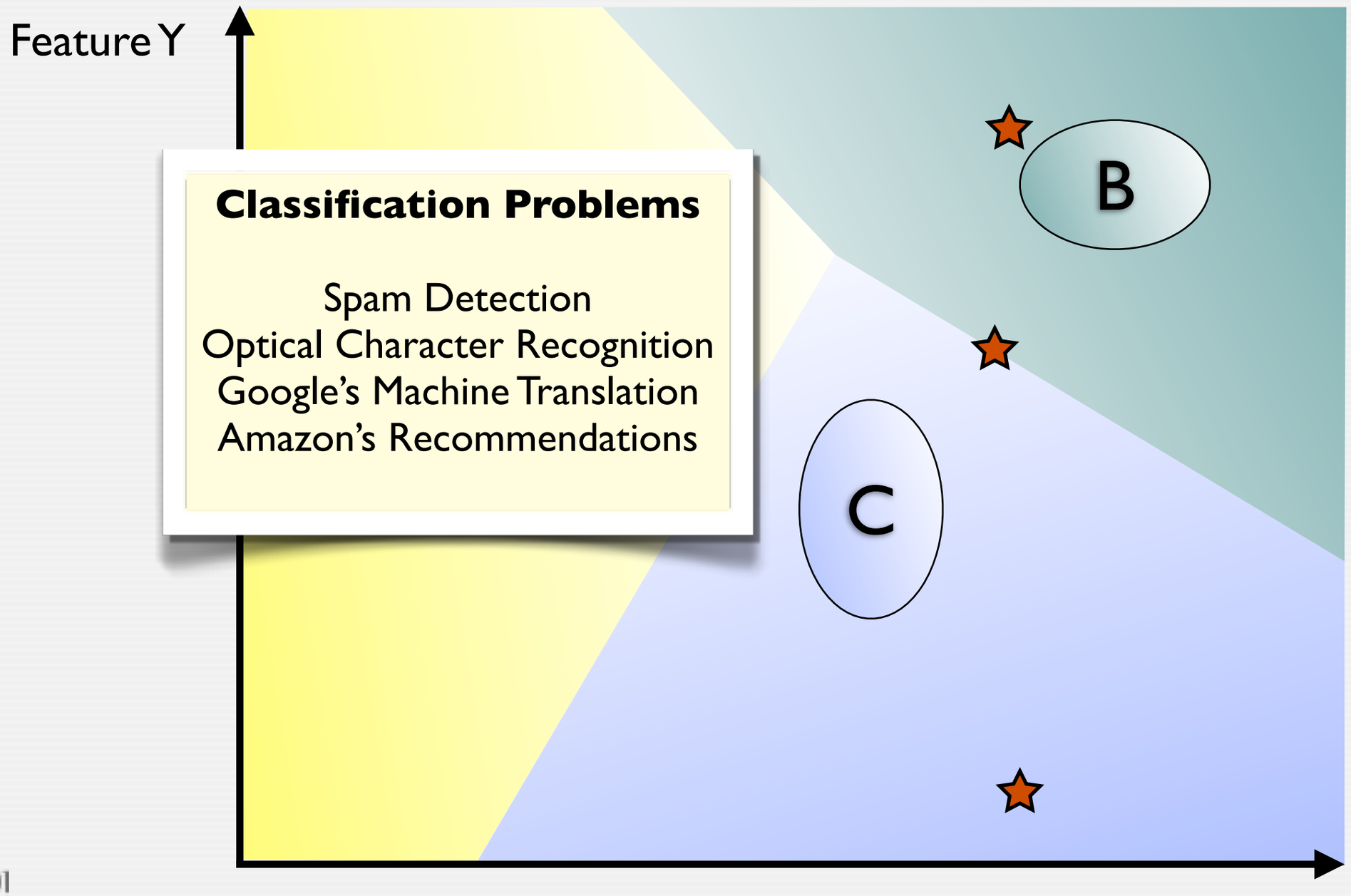
Classification Problem



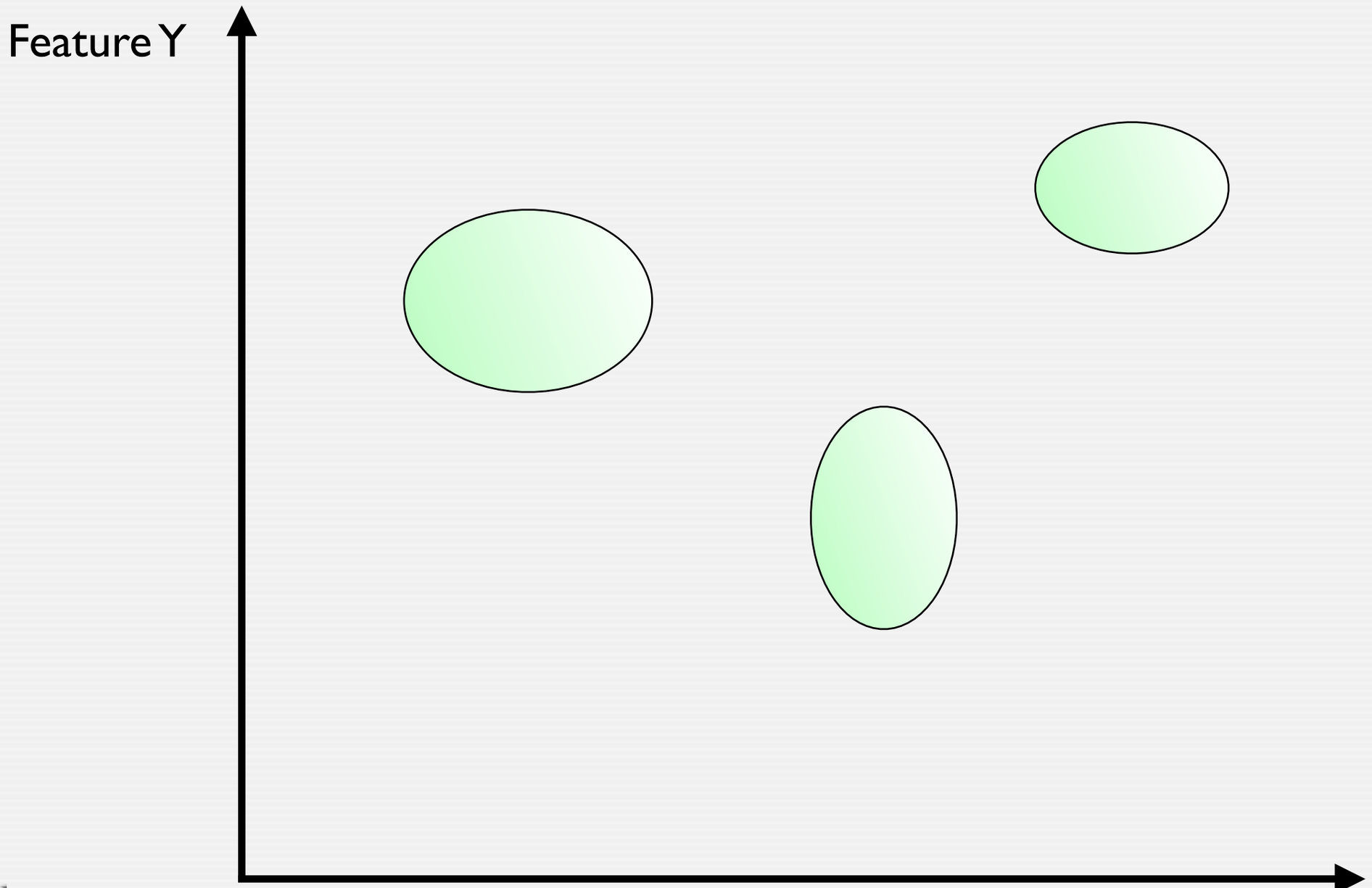
Classification Problem



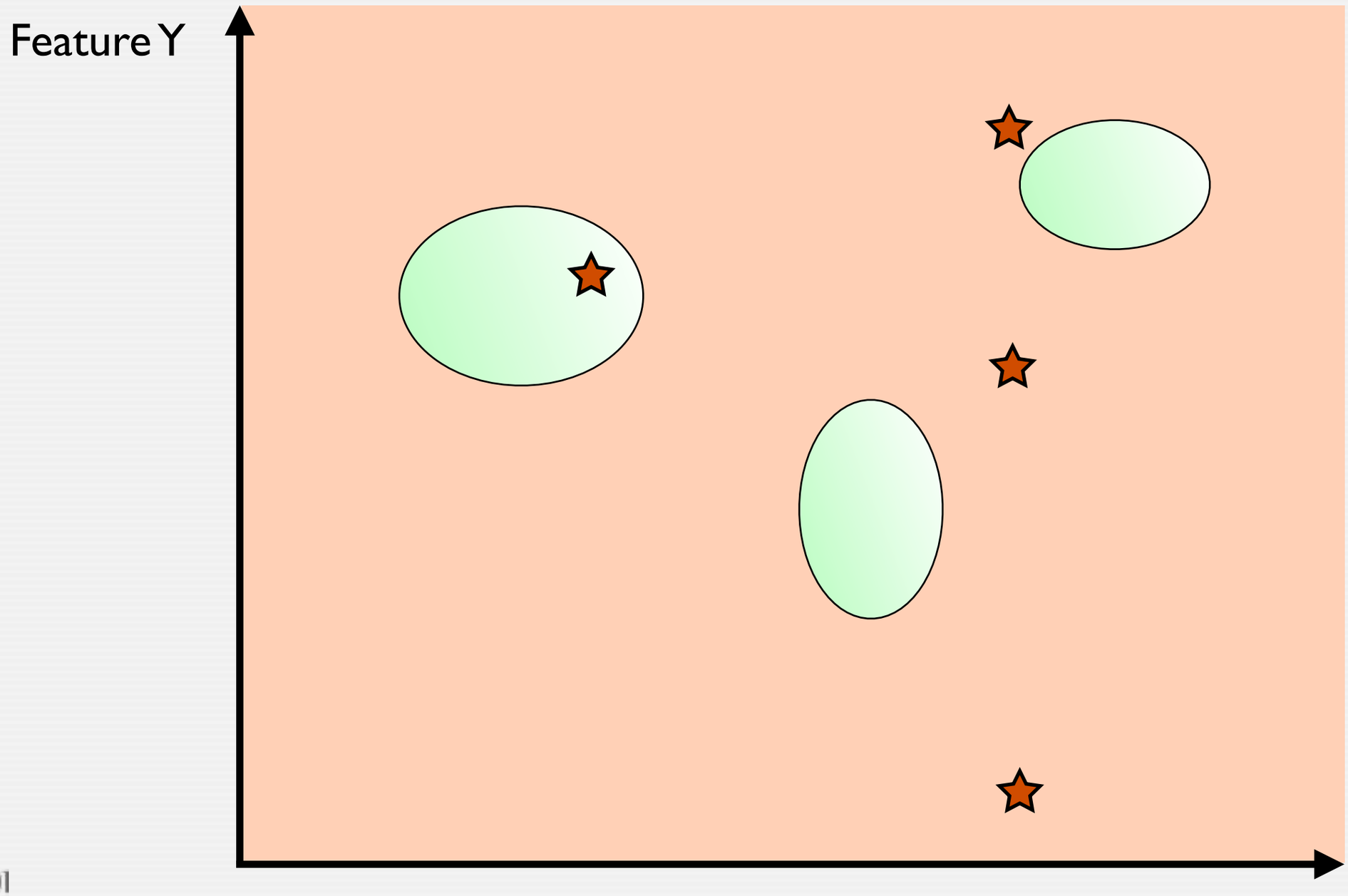
Classification Problem



Outlier Detection



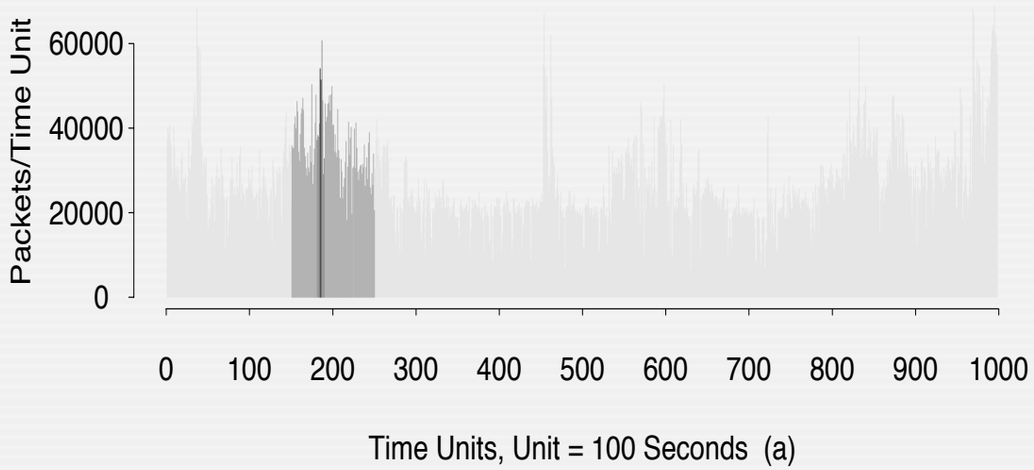
Outlier Detection



Outlier Detection

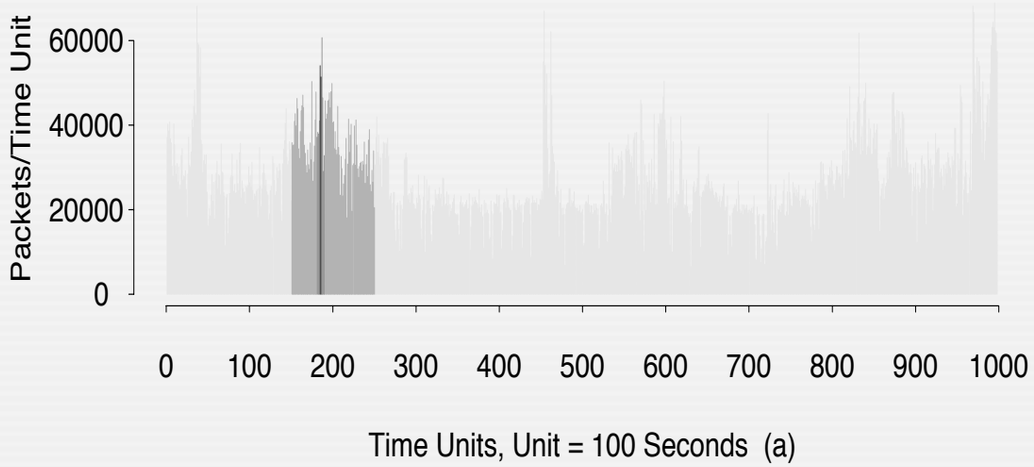
- Assumes a *Closed World:*
  - Specify only positive examples.
  - Adopt standing assumption that the rest is negative.
- Real-life problems rarely involve "closed" worlds.
  - One needs to cover all positive cases to avoid misclassifications.
- Can be used successfully if the model is "good enough".
  - Feature space is of low dimensionality and/or variability.
  - Mistakes are cheap.
  - Examples: fraud detection (credit cards, insurances); image analysis.
- Tends to be hard to do for intrusion detection
  - Network activity is extremely diverse at all levels of the protocol stack.
  - ... and that's already without any malicious activity.

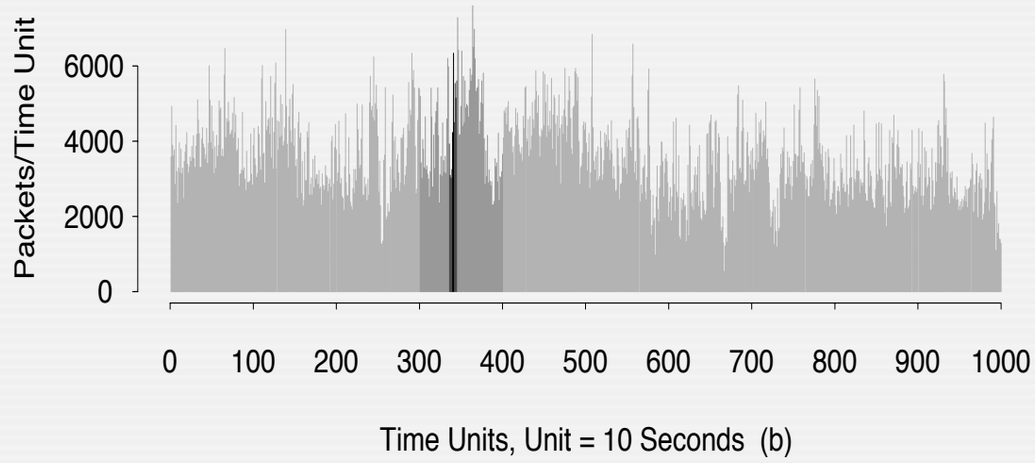
Self-Similarity of Ethernet Traffic



Source: LeLand et al. 1995

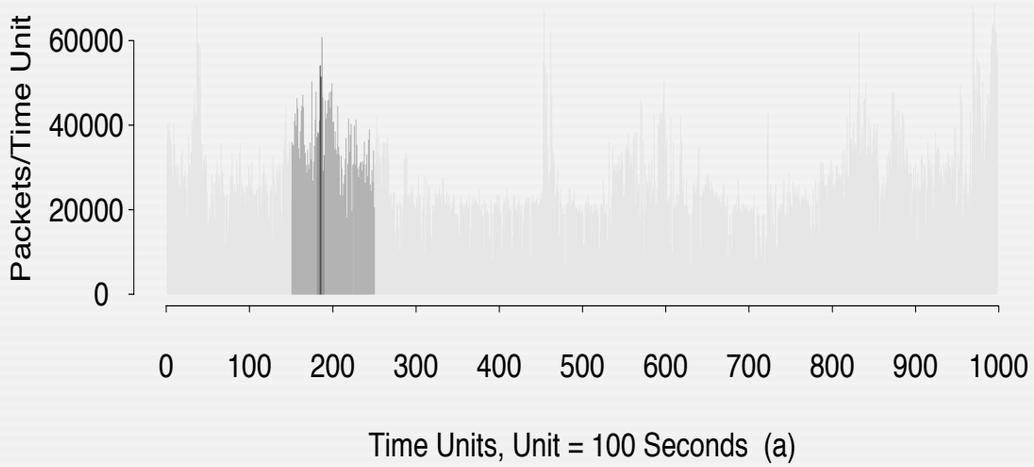
Self-Similarity of Ethernet Traffic

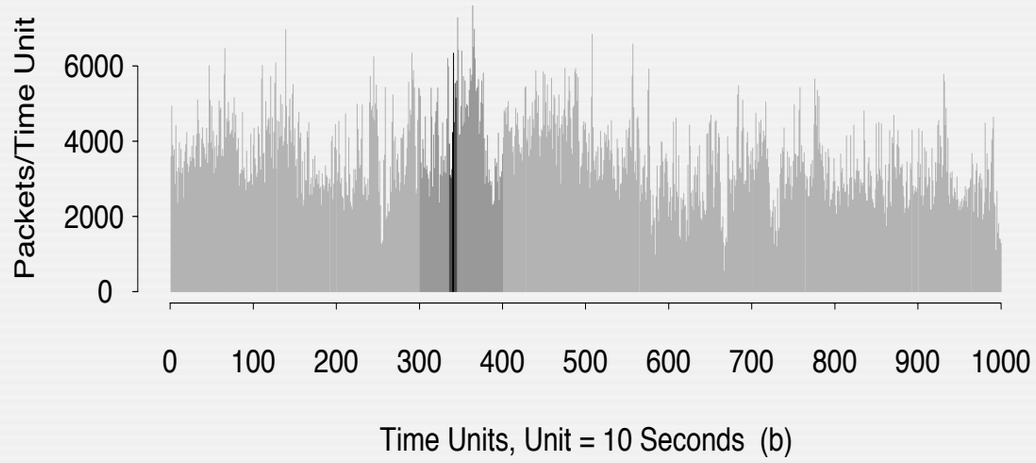


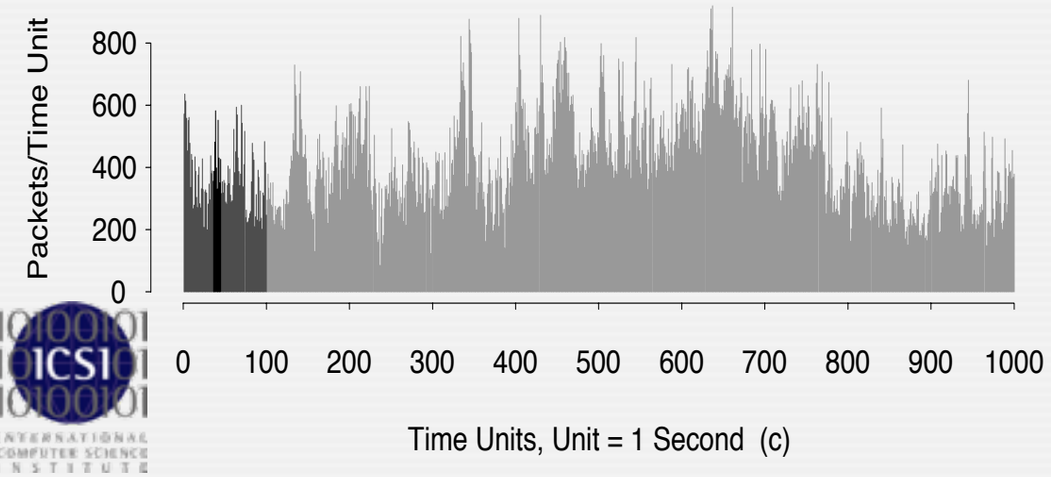


Source: LeLand et al. 1995

Self-Similarity of Ethernet Traffic

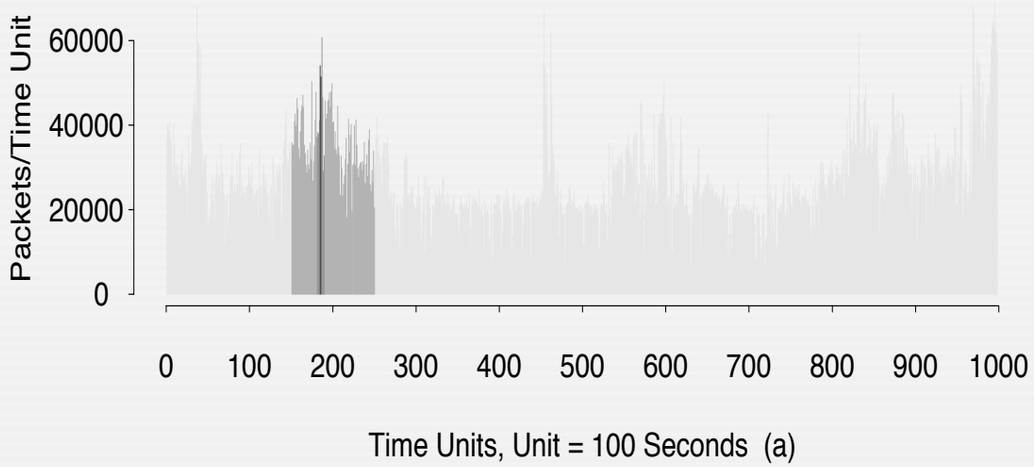


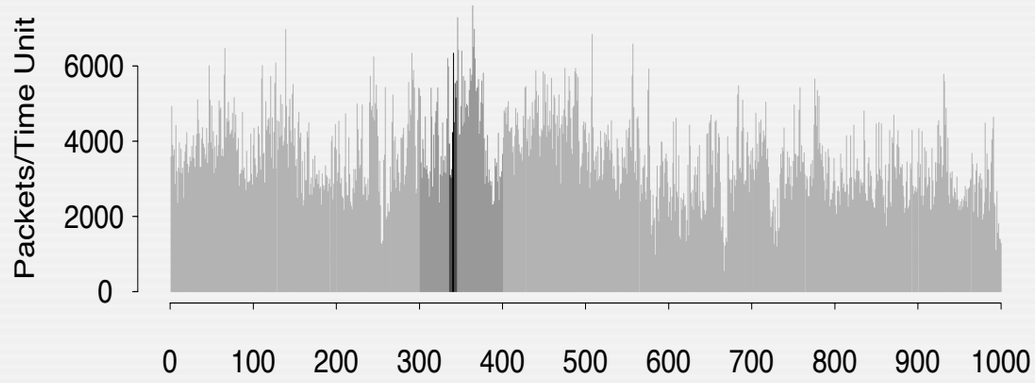


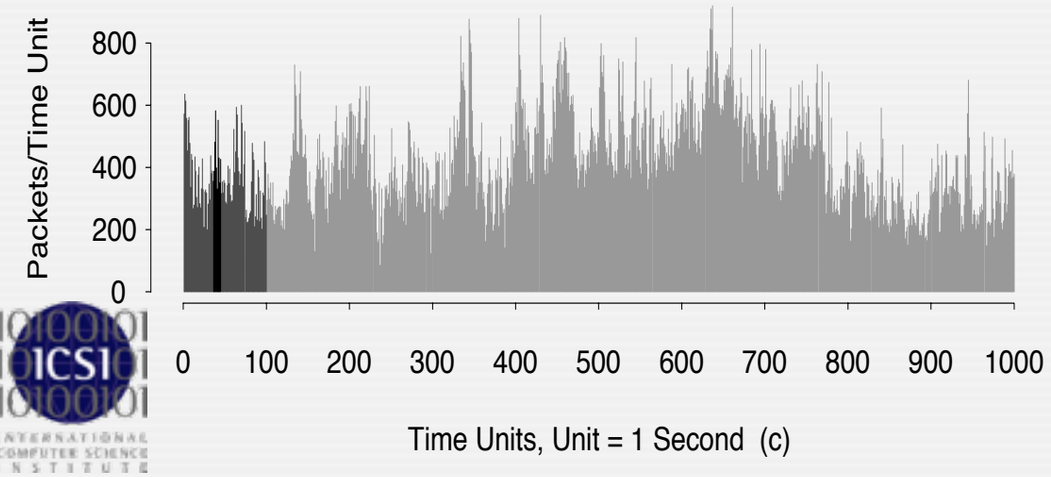


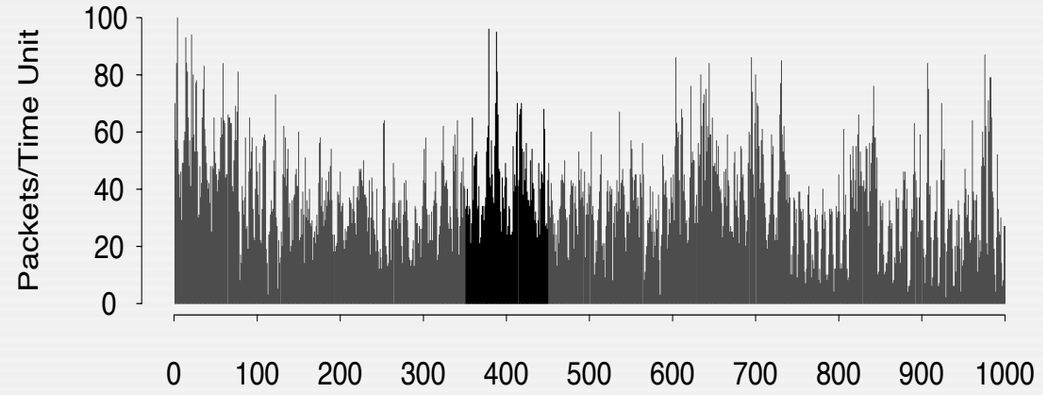
Source: LeLand et al. 1995

Self-Similarity of Ethernet Traffic





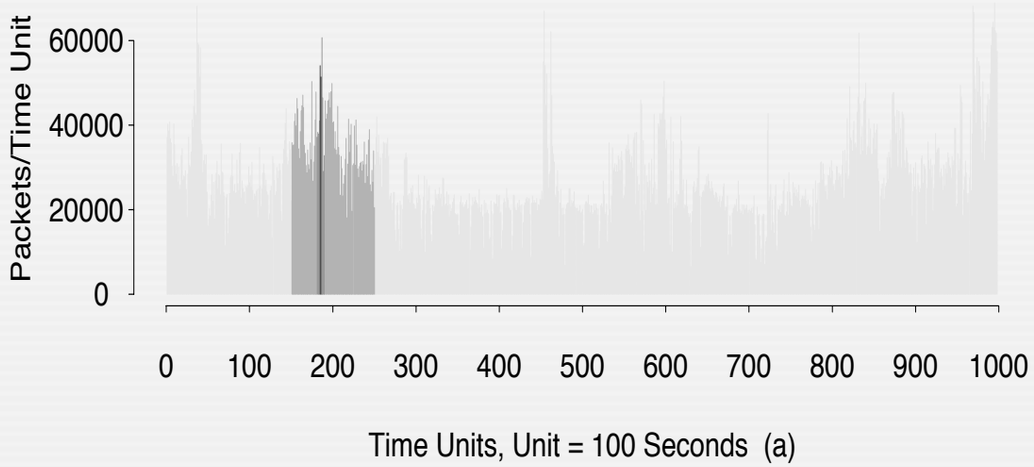


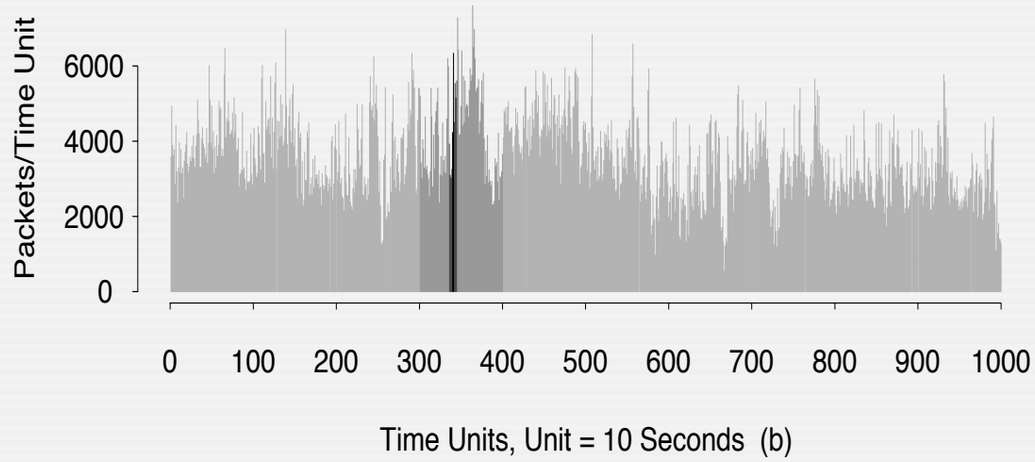


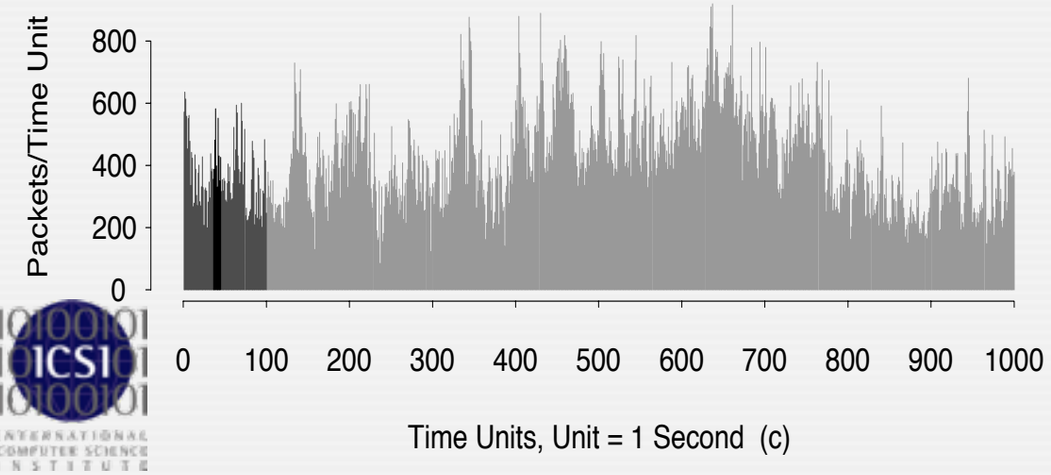
Time Units, Unit = 0.1 Second (d)

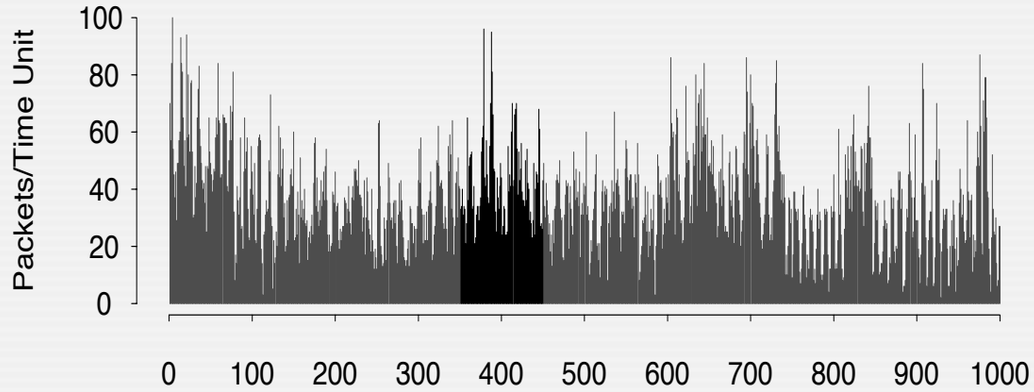
Source: LeLand et al. 1995

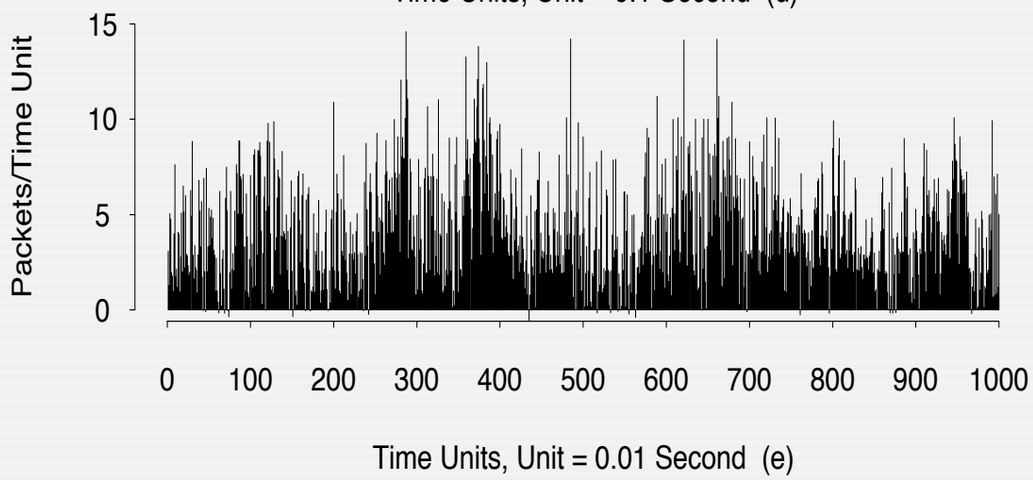
Self-Similarity of Ethernet Traffic





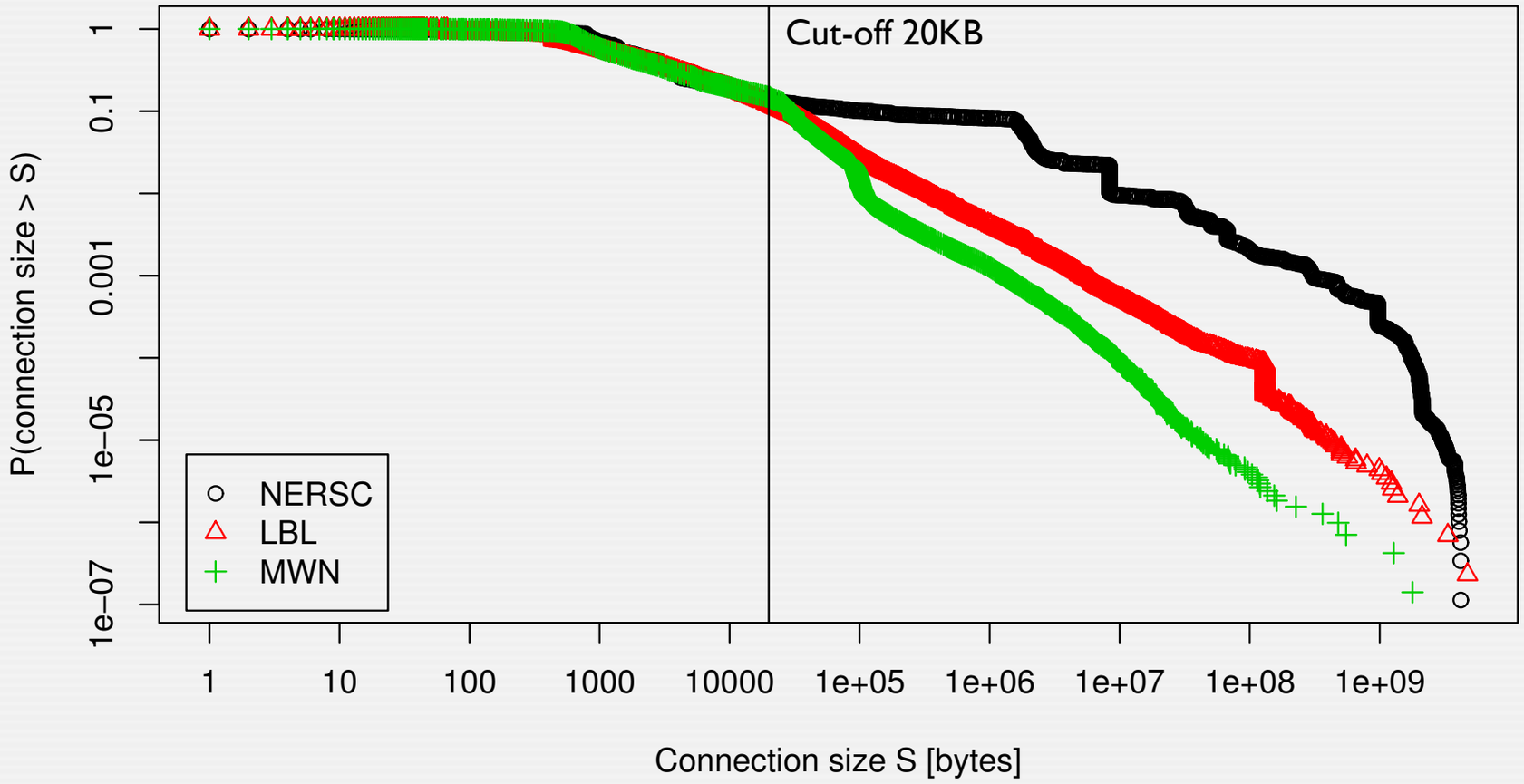






Source: LeLand et al. 1995

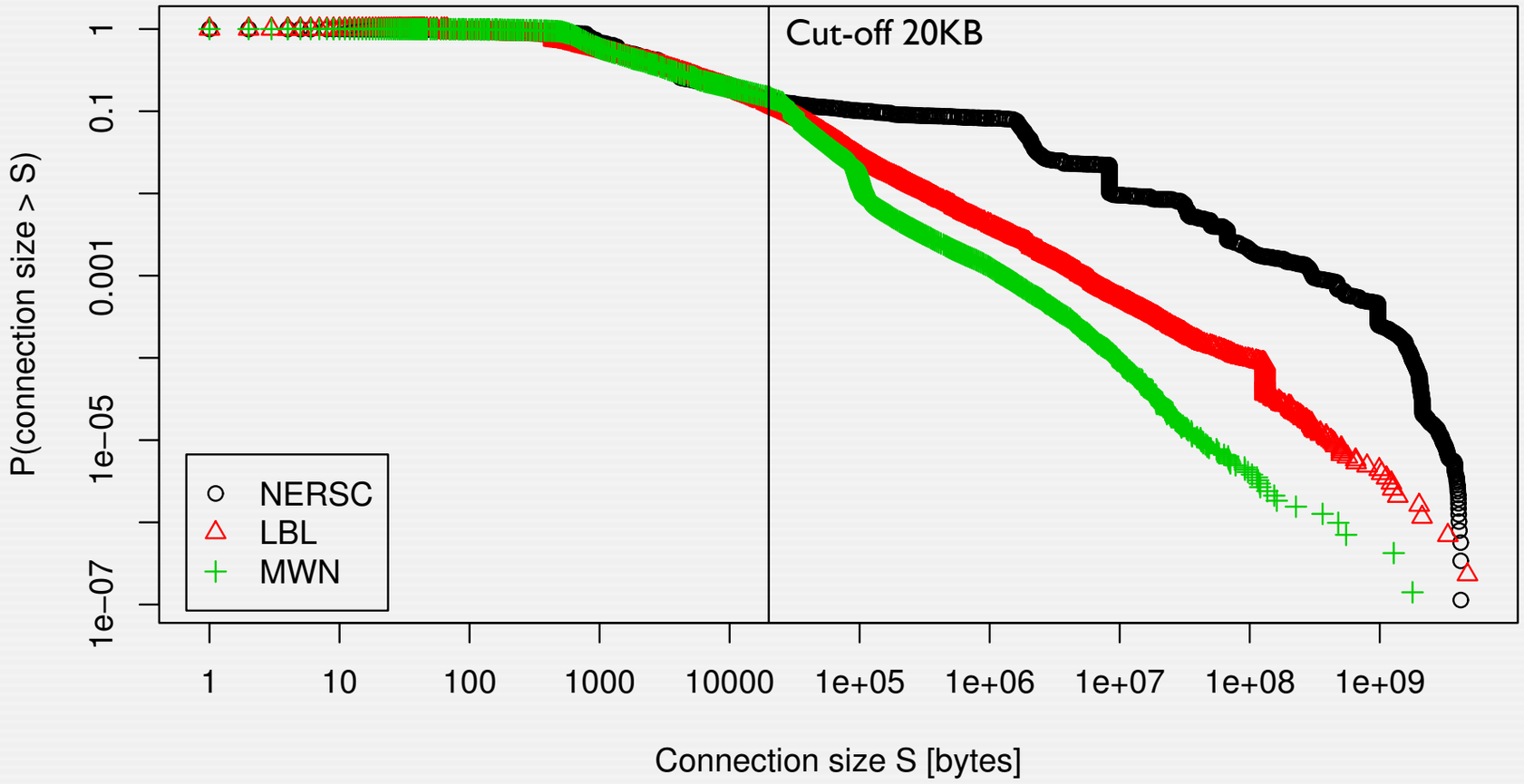
Heavy Tails



Site	Conns > 20KB	%Bytes
MWN	15%	87%
LBL	12%	96%
NERSC	14%	99.86%

Source: Kornexl 2005

Heavy Tails

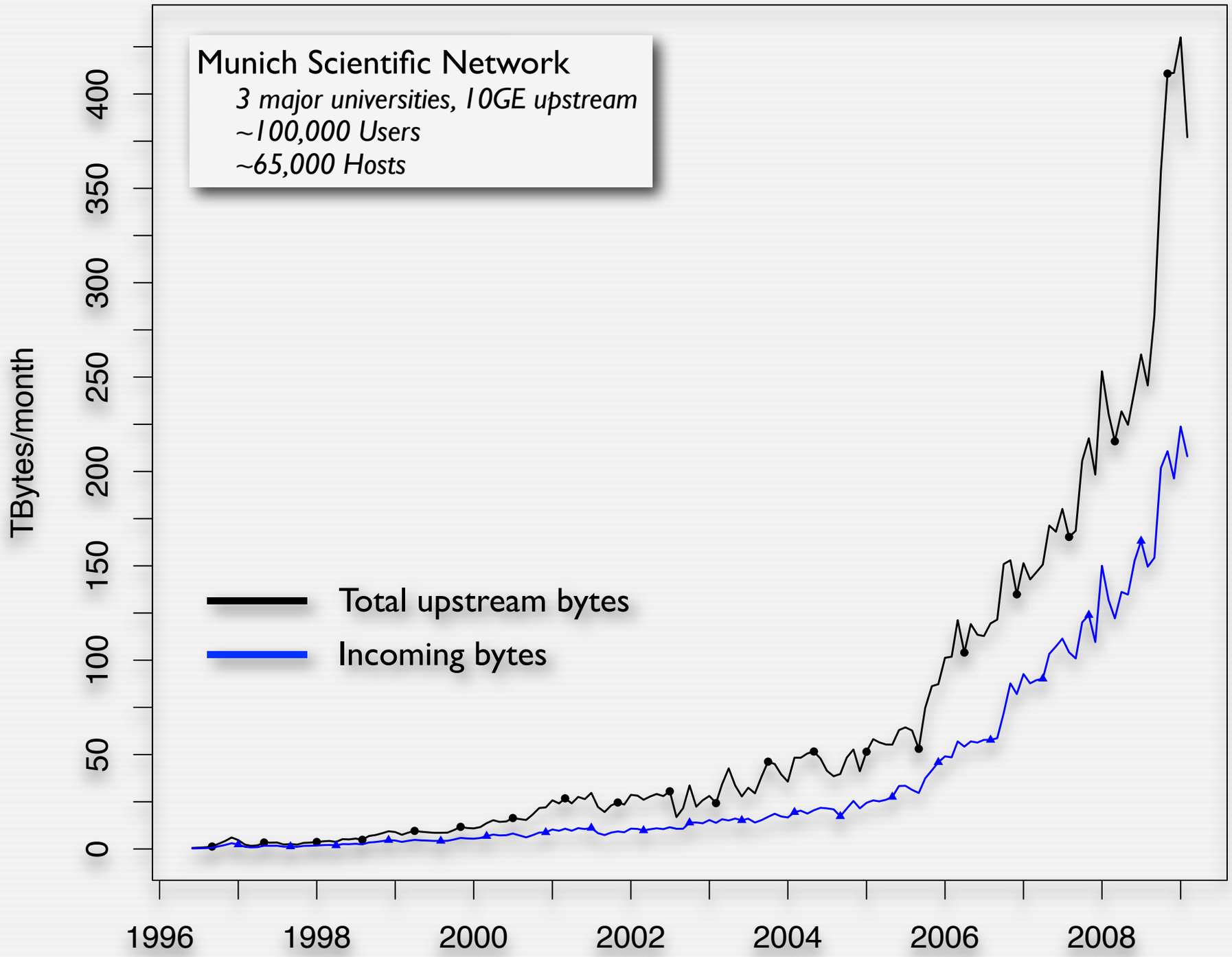


Site	Conns > 20KB	%Bytes
MWN	15%	87%
LBL	12%	96%
NERSC	14%	99.86%

Self-similarity/heavy-tails lead to extreme bursts.

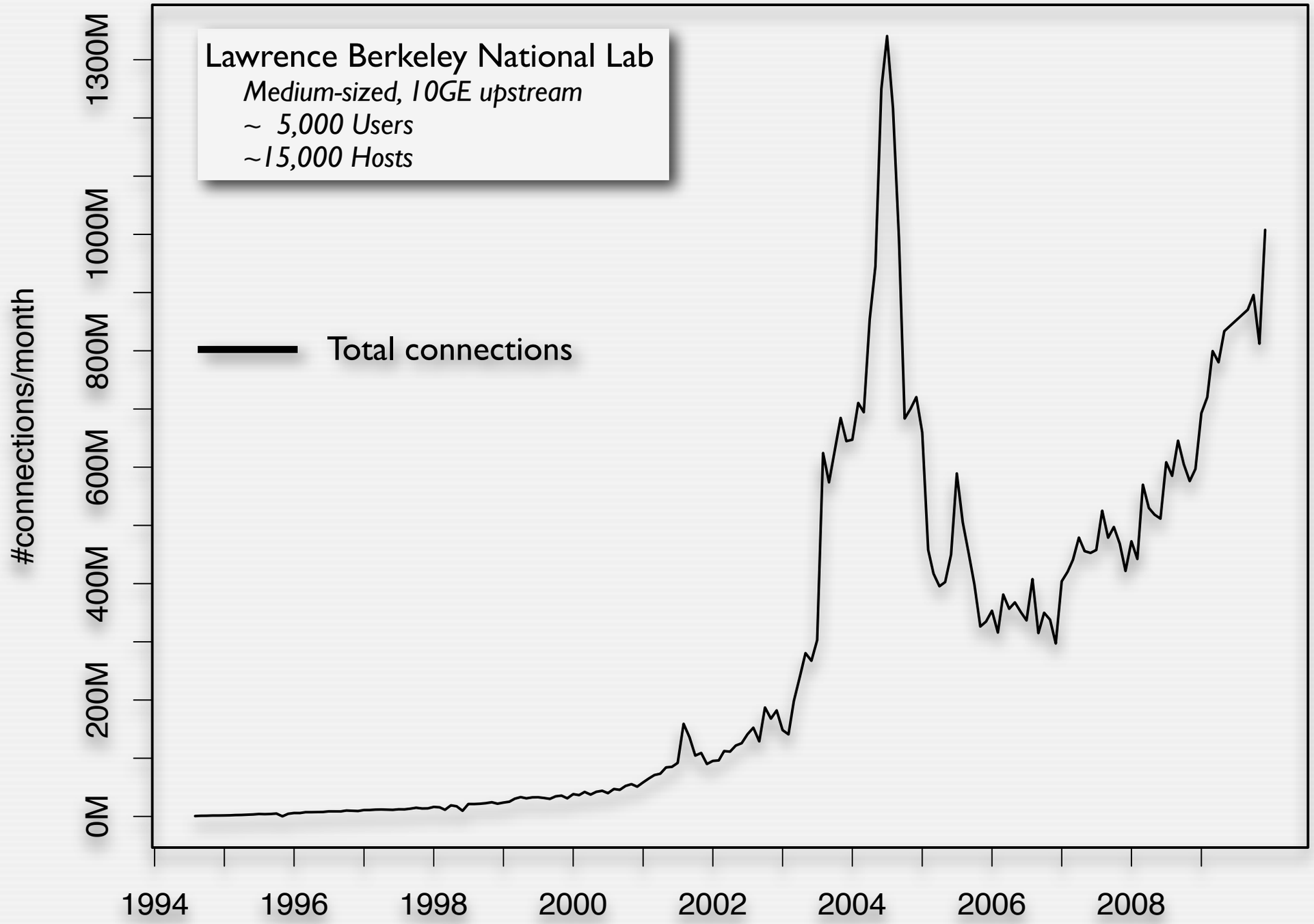
Source: Kornexl 2005

A Moving Target ...



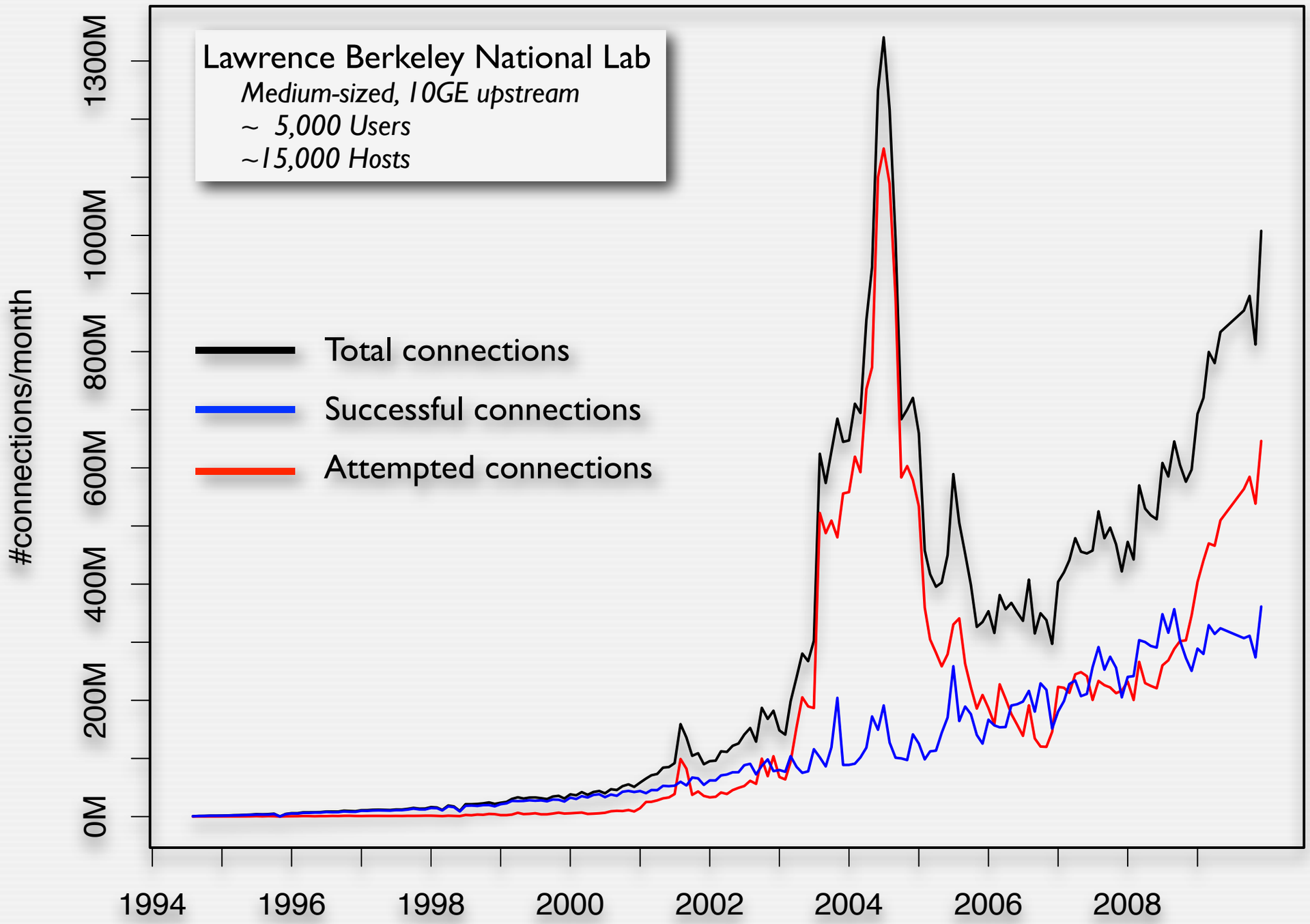
Data: Leibniz-Rechenzentrum, München

Internet Traffic: Connections



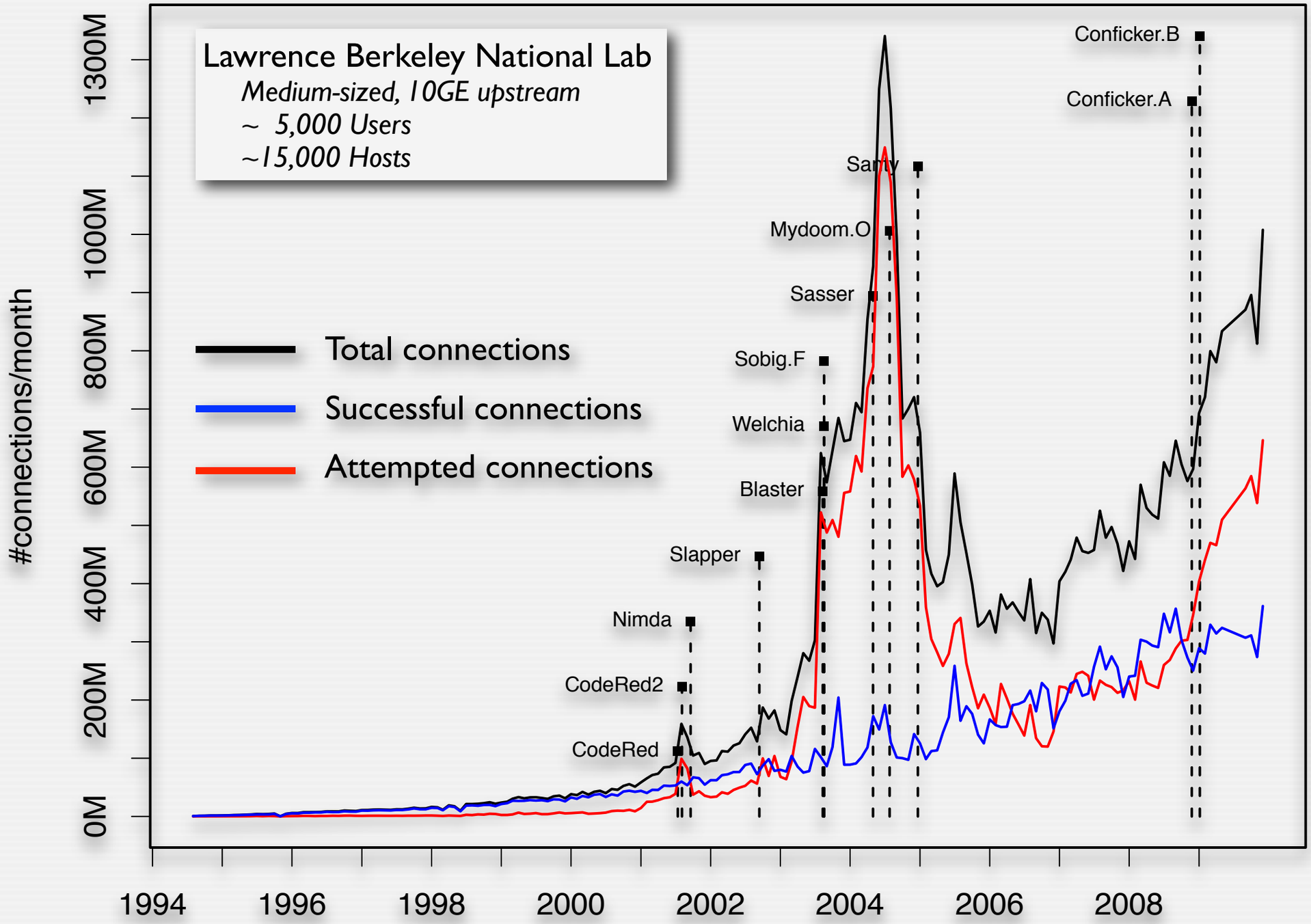
Data: Lawrence Berkeley National Lab

Internet Traffic: Connections



Data: Lawrence Berkeley National Lab

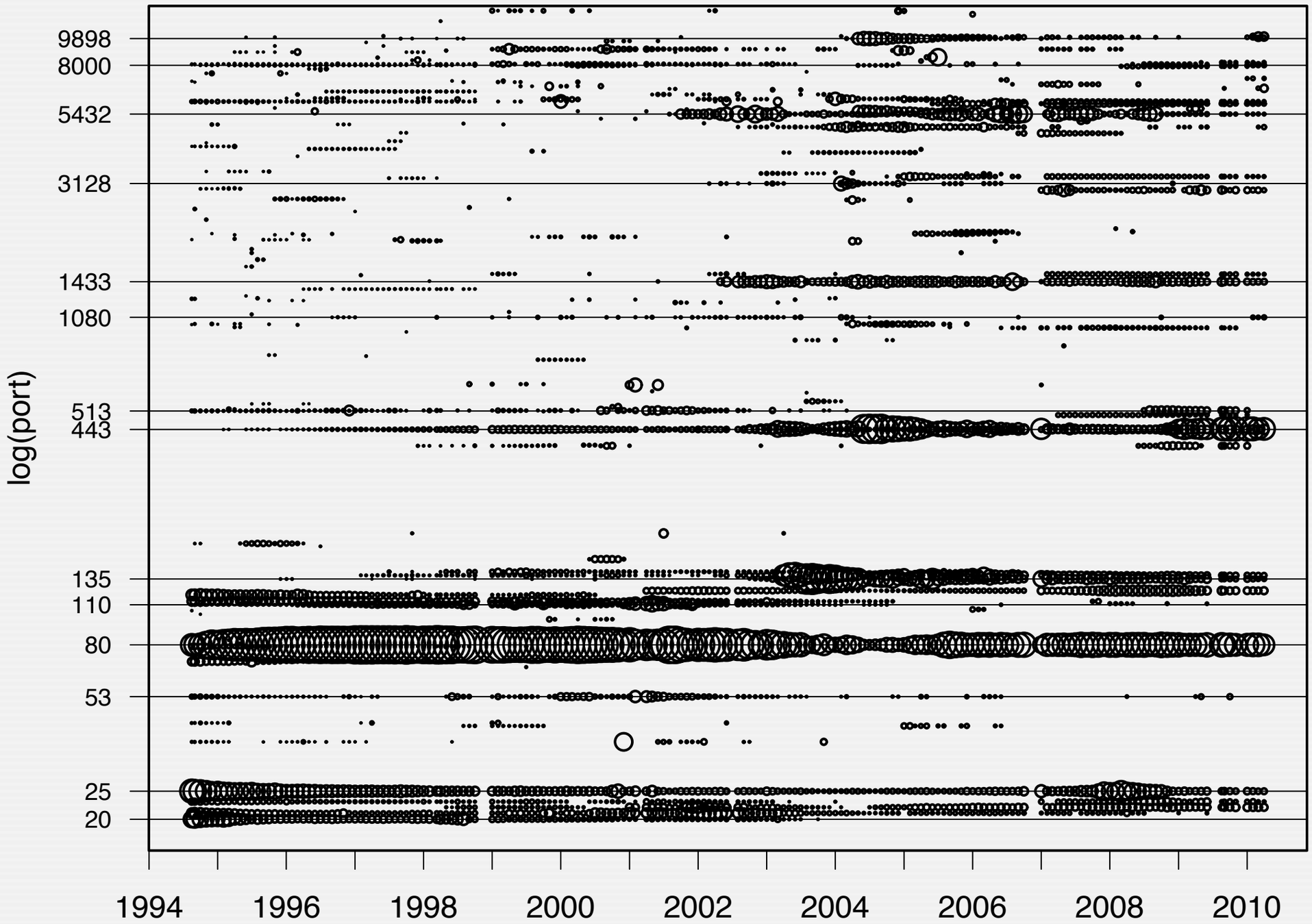
Internet Traffic: Connections



Data: Lawrence Berkeley National Lab

A Moving Target ... (2)

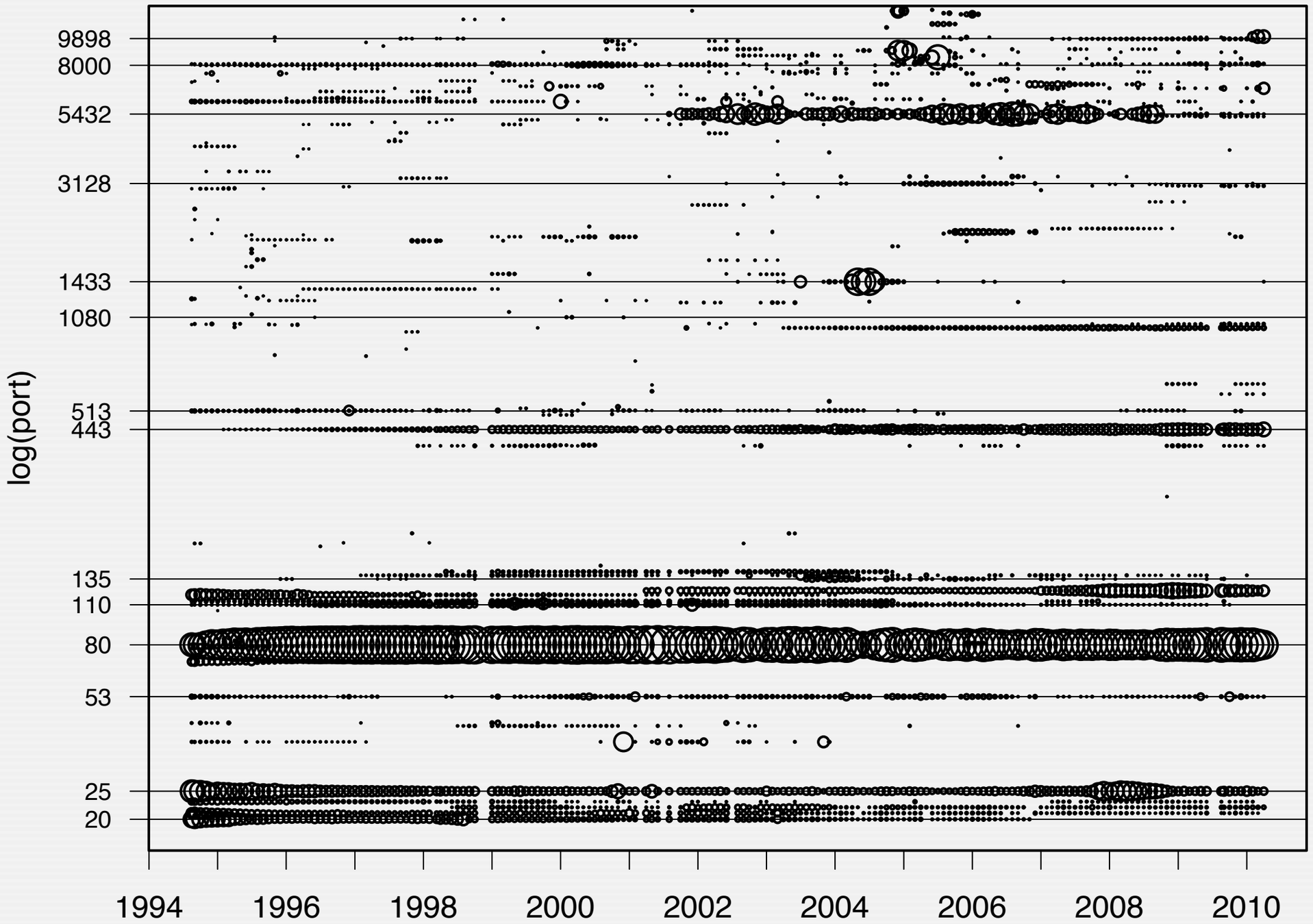
All connections.



Data: Lawrence Berkeley National Lab

A Moving Target ... (2)

Successful connections.



Data: Lawrence Berkeley National Lab

One Day of Crud at ICSI

Postel's Law: *Be strict in what you send and liberal in what you accept ...*

active-connection-reuse	DNS-label-len-gt-pkt	HTTP-chunked-multipart	possible-split-routing
bad-Ident-reply	DNS-label-too-long	HTTP-version-mismatch	SYN-after-close
bad-RPC	DNS-RR-length-mismatch	illegal-%-at-end-of-URI	SYN-after-reset
bad-SYN-ack	DNS-RR-unknown-type	inappropriate-FIN	SYN-inside-connection
bad-TCP-header-len	DNS-truncated-answer	IRC-invalid-line	SYN-seq-jump
base64-illegal-encoding	DNS-len-lt-hdr-len	line-terminated-with-single-CR	truncated-NTP
connection-originator-SYN-ack	DNS-truncated-RR-rdlength	malformed-SSH-identification	unescaped-%-in-URI
data-after-reset	double-%-in-URI	no-login-prompt	unescaped-special-URI-char
data-before-established	excess-RPC	NUL-in-line	unmatched-HTTP-reply
too-many-DNS-queries	FIN-advanced-last-seq	POP3-server-sending-client-commands	window-recision
DNS-label-forward-compress-	fragment-with-DF		***155K in total!***

Is There a Stable Notion of *Normal?*

- Internet traffic is composed of *many* individual sessions.
  - Leads to enormous variety and unpredictable behavior.
- Complex distributions of features:
  - Self-similarity, heavy tails, long-range dependance.
  - Constantly changing.
  - Incessant background noise and tons of crud.
- Observable on all layers of the protocol stack.
- In general, it's pretty much impossible to define "normal".
  - Huge fluctuations are *normal* and *expected* short-term.
  - No attacker needed for that!

Are Outliers Attacks?

- Implicit assumption that anomaly detectors make:

Outliers are malicious!

- With such diversity, that can be hard to justify.
  - Most familiar with the matter will say "anomaly detection doesn't report attacks."
  - Instead, it's up to the operator to investigate which outliers are indeed malicious.
- That leads to a *semantic gap*.
  - Disconnect between what the system reports and what the operator wants.
  - Root cause for the common complaint of "too many false positives".
- An operator must be able to understand the alarms.
  - If it may or may not be an attack, there must a means to separate the two.
  - This is certainly hard: the system doesn't know the cause.
  - But still, it's not helpful to just ignore the issue.

Relating Features To Semantics

- Key question: What *can* our features tell us?
  - Do packet arrival times tell us something about SQL injection?
  - Do NetFlow records allow us to find inappropriate content?
  - What are the right features to learn how SSNs look like?
- Need to consider a site's security policy as well.
  - What *is* tolerable usage of P2P systems?
  - What *is* appropriate content?
- There are striking examples of how much more information a data set might contain than expected.
- But one needs to exploit structural knowledge.
  - Hard to see a classifier just "learning" peculiar activity.

Every Mistake Is Expensive

- Each false alert costs scarce analyst time.
  - Go through log files, inspect system, talk to users, etc.
  - "Trains" the operator to mistrust future alarms.
- In other domains, errors tend to be cheap.
  - Wrong recommendation from Amazon? Not a big deal.
    Greg Linden: " ... *guess work. Our error rate will always be high.*"
  - Letter misclassified by an OCR system? Spell-checker.
  - Machine translation? Have you tried it?
  - Spam detection? Lopsided tuning.
- What error rate can we afford with an IDS?

Base-Rate Fallacy

- Doctor performs a disease test that is 99% accurate.
  - If in a group all have the disease, the test reports 99% as positive.
  - If in a group none has the disease, the test reports 99% as negative.
- Bad news: your test comes back *positive*.
  - However, your doctor says that overall only 1 in 10000 people has the disease
- So, what's the likelihood that you have it?

Source: Axelsson 1999

Base-Rate Fallacy

- Doctor performs a disease test that is 99% accurate.
  - If in a group all have the disease, the test reports 99% as positive.
  - If in a group none has the disease, the test reports 99% as negative.
- Bad news: your test comes back *positive*.
  - However, your doctor says that overall only 1 in 10000 people has the disease
- So, what's the likelihood that you have it?

$$P(P|S) = \frac{P(P)*P(S|P)}{P(P)}$$

Bayes' Theorem

Source: Axelsson 1999

Base-Rate Fallacy

- Doctor performs a disease test that is 99% accurate.
  - If in a group all have the disease, the test reports 99% as positive.
  - If in a group none has the disease, the test reports 99% as negative.
- Bad news: your test comes back *positive*.
  - However, your doctor says that overall only 1 in 10000 people has the disease
- So, what's the likelihood that you have it?

$$P(P|S) = \frac{P(P)*P(S|P)}{P(P)}$$

Bayes' Theorem

$$P(S|P) = \frac{P(S)*P(P|S)}{P(S)*P(P|S)+P(\neg S)*P(P|\neg S)}$$

Source: Axelsson 1999

Base-Rate Fallacy

- Doctor performs a disease test that is 99% accurate.
  - If in a group all have the disease, the test reports 99% as positive.
  - If in a group none has the disease, the test reports 99% as negative.
- Bad news: your test comes back *positive*.
  - However, your doctor says that overall only 1 in 10000 people has the disease
- So, what's the likelihood that you have it?

Bayes' Theorem

$$P(P|S) = \frac{P(P)*P(S|P)}{P(P)}$$

$$P(S|P) = \frac{P(S)*P(P|S)}{P(S)*P(P|S)+P(\neg S)*P(P|\neg S)}$$

$$= \frac{\frac{1}{10000}*0.99}{\frac{1}{10000}*0.99+(1-\frac{1}{10000})*0.01} \approx 1\%$$

Source: Axelsson 1999

Bayesian Detection Rate

1M audit records / day
2 intrusions / day
10 records / intrusion

$$P(I) = 1/\frac{1*10^6}{2*10} = 2*10^{-5}$$

Source: Axelsson 1999

Bayesian Detection Rate

1M audit records / day
2 intrusions / day
10 records / intrusion

$$P(I) = 1/\frac{1*10^6}{2*10} = 2*10^{-5}$$

$$P(I|A) = \frac{P(I)*P(A|I)}{P(I)*P(A|I)+P(\neg I)*P(A|\neg I)}$$

Source: Axelsson 1999

Bayesian Detection Rate

1M audit records / day
2 intrusions / day
10 records / intrusion

$$P(I) = 1/\frac{1*10^6}{2*10} = 2*10^{-5}$$

$$P(I|A) = \frac{P(I)*P(A|I)}{P(I)*P(A|I)+P(\neg I)*P(A|\neg I)}$$

$$= \frac{2*10^{-5}*P(A|I)}{2*10^{-5}*P(A|I)+0.99998*P(A|\neg I)}$$

Source: Axelsson 1999

Bayesian Detection Rate

1M audit records / day
2 intrusions / day
10 records / intrusion

$$P(I) = 1/\frac{1*10^6}{2*10} = 2*10^{-5}$$

$$P(I|A) = \frac{P(I)*P(A|I)}{P(I)*P(A|I)+P(\neg I)*P(A|\neg I)}$$

Detection rate

$$= \frac{2*10^{-5}*P(A|I)}{2*10^{-5}*P(A|I)+0.99998*P(A|\neg I)}$$

Source: Axelsson 1999

Bayesian Detection Rate

1M audit records / day
2 intrusions / day
10 records / intrusion

$$P(I) = 1/\frac{1*10^6}{2*10} = 2*10^{-5}$$

$$P(I|A) = \frac{P(I)*P(A|I)}{P(I)*P(A|I)+P(\neg I)*P(A|\neg I)}$$

Detection rate

$$= \frac{2*10^{-5}*P(A|I)}{2*10^{-5}*P(A|I)+0.99998*P(A|\neg I)}$$

False alarm rate

Source: Axelsson 1999

Bayesian Detection Rate

1M audit records / day
2 intrusions / day
10 records / intrusion

$$P(I) = 1/\frac{1*10^6}{2*10} = 2*10^{-5}$$

$$P(I|A) = \frac{P(I)*P(A|I)}{P(I)*P(A|I)+P(\neg I)*P(A|\neg I)}$$

Detection rate

$$= \frac{2*10^{-5}*P(A|I)}{2*10^{-5}*P(A|I)+0.99998*P(A|\neg I)}$$

False alarm rate

Even with perfect *detection*, the false alarm rate must be on the order of 10^{-5} to get 2/3 of all alarms correct.

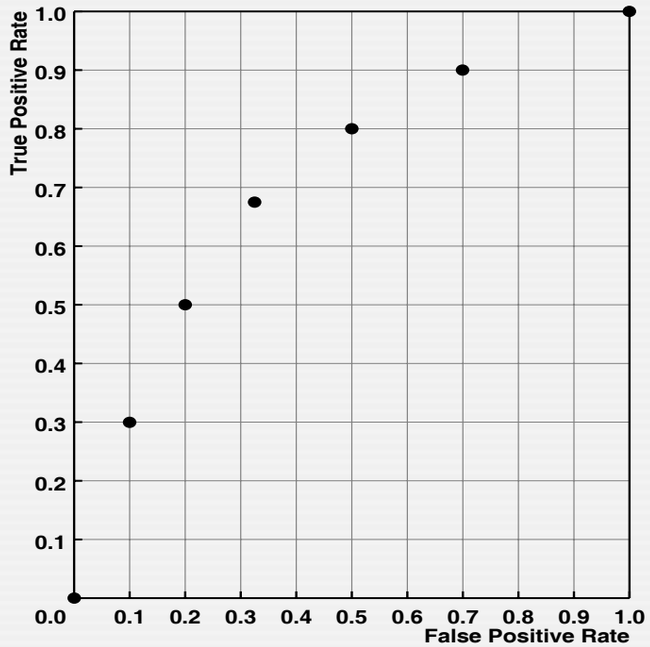
Source: Axelsson 1999

How Do We Measure Performance?

Observer Decision	Truth: Signal + Noise	Truth: Noise alone
Yes	True-positive Rate a.k.a. **Hit Rate:** $\frac{\textbf{\# Hits}}{\text{Total \# Signal Events}}$	False-positive Rate a.k.a. **False-alarm Rate:** $\frac{\textbf{\# False Alarms}}{\text{Total \# Noise Events}}$
No	False-negative Rate a.k.a. **Miss Rate:** $\frac{\textbf{1 - \# Hits}}{\text{Total \# Signal Events}}$	True-negative Rate a.k.a. **Correct-rej. Rate:** $\frac{\textbf{1 - \# False Alarms}}{\text{Total \# Noise Events}}$

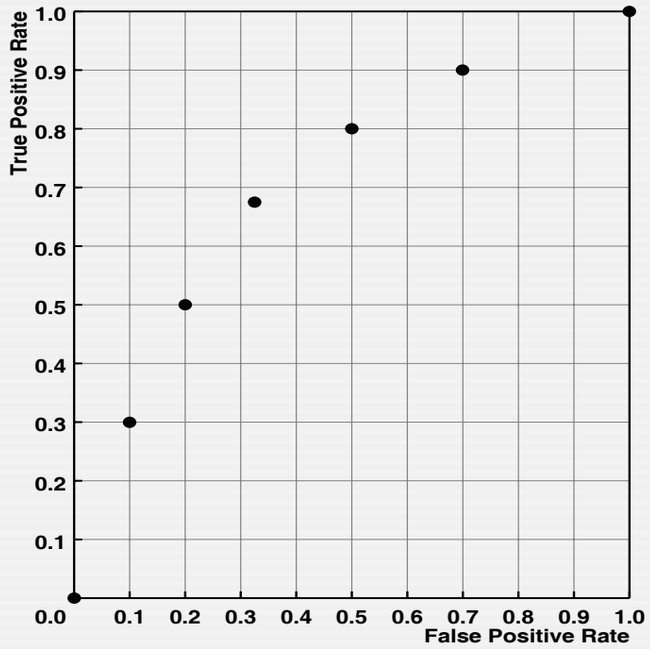
Source: Maxion/Roberts 2004

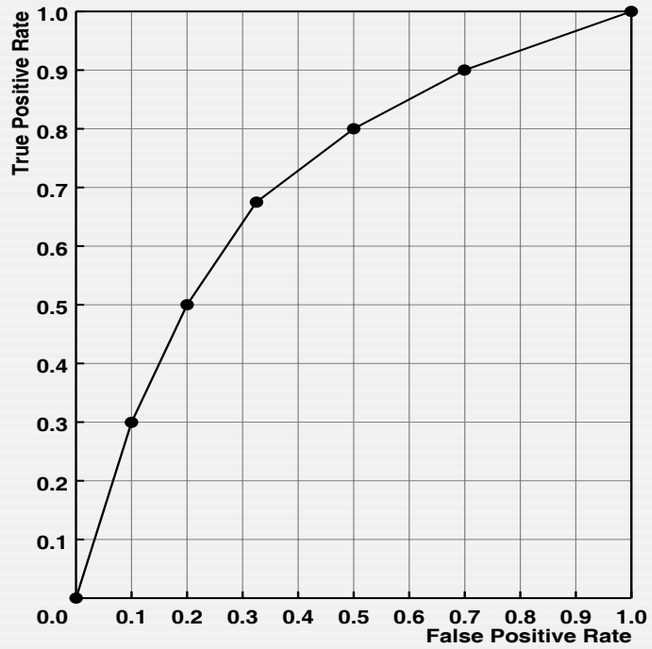
Receiver Operating Characteristic



Source: Maxion/Roberts 2004

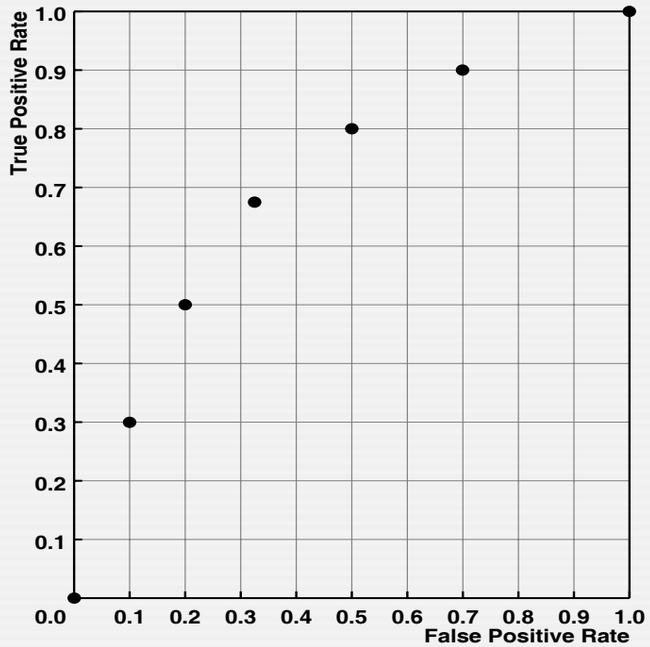
Receiver Operating Characteristic

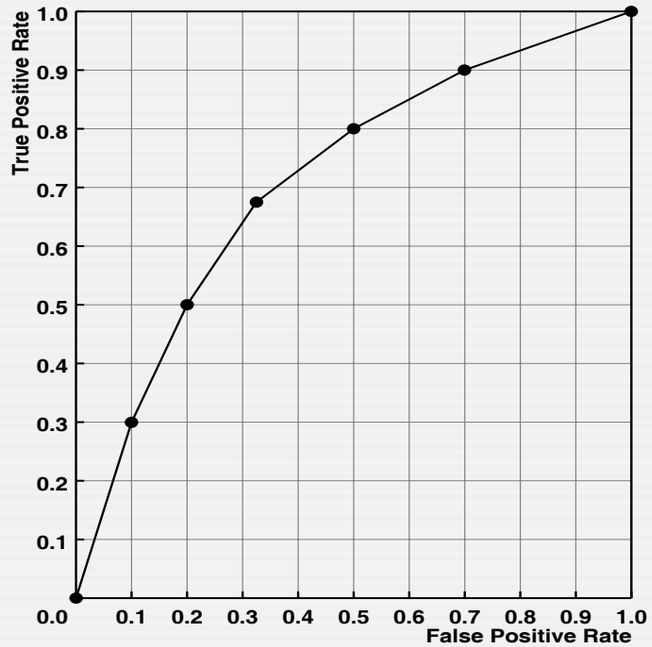


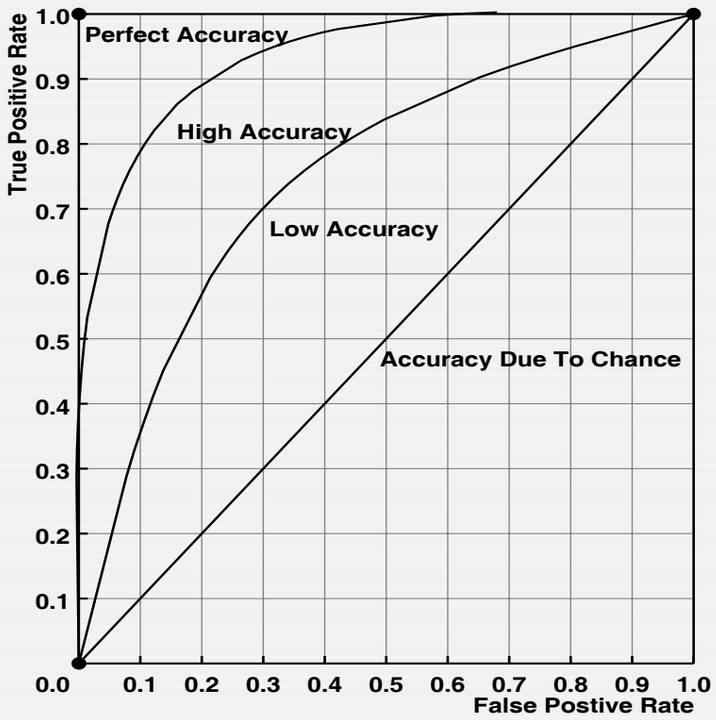


Source: Maxion/Roberts 2004

Receiver Operating Characteristic

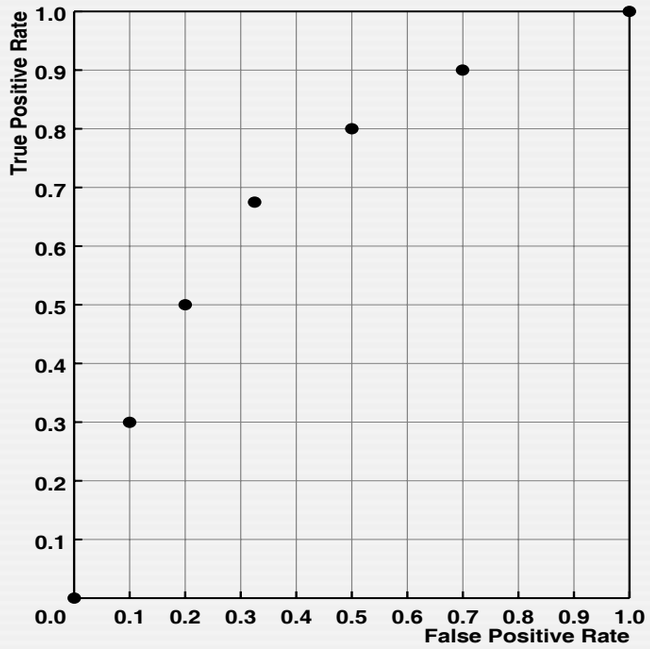


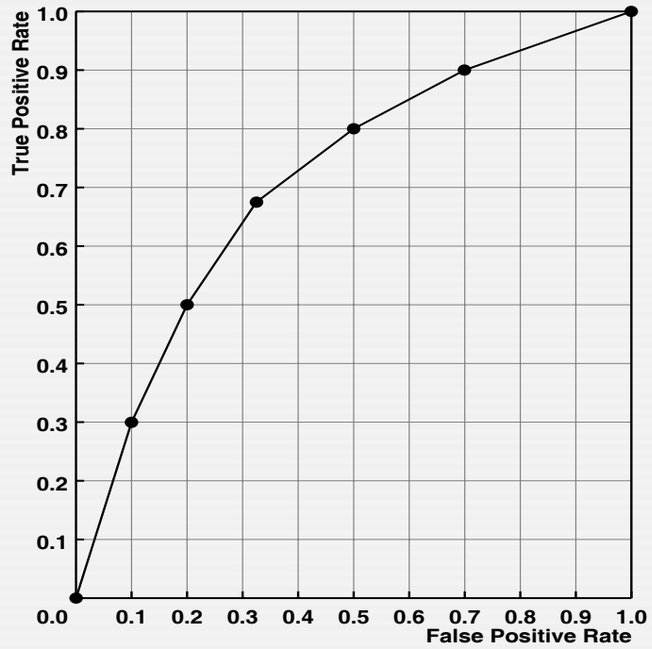


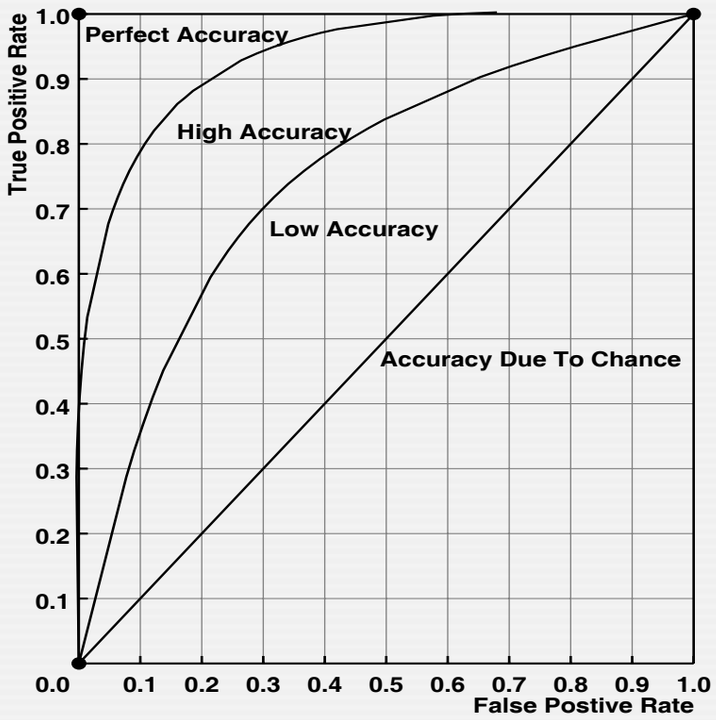


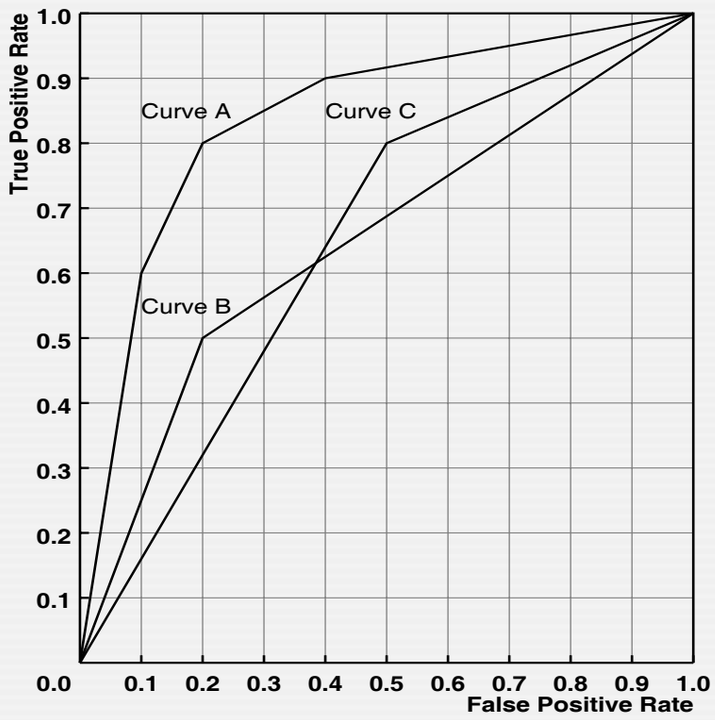
Source: Maxion/Roberts 2004

Receiver Operating Characteristic









Source: Maxion/Roberts 2004

So Much for the Theory ...

The notion of false positive/negative rate is quite fuzzy.

Unit of analysis

How do we count "*noise events*" *and* "*signal events*"?

Ground truth

How do we know/define what is a true attack?

Evaluation data set

How do we know that it is representative?

All these notions depend on detector and environment, making it extremely hard to fairly compare systems with each other.

A Standard Corpus For Evaluation?

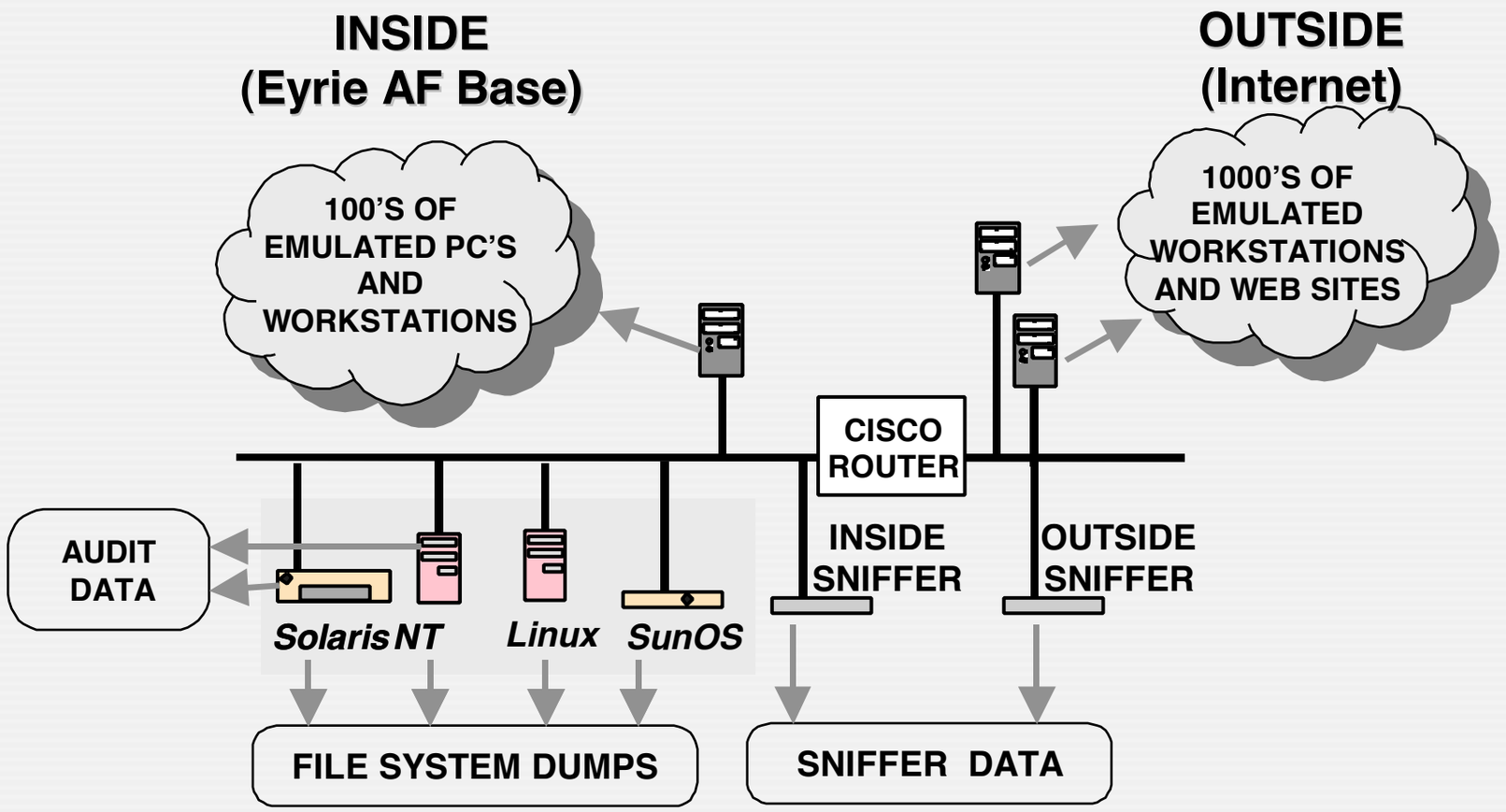
DARPA/Lincoln Labs & KDD Cup Data Sets

An attempt at providing the community with public data sets for fair NIDS evaluation

A Standard Corpus For Evaluation?

DARPA/Lincoln Labs & KDD Cup Data Sets

An attempt at providing the community with public data sets for fair NIDS evaluation

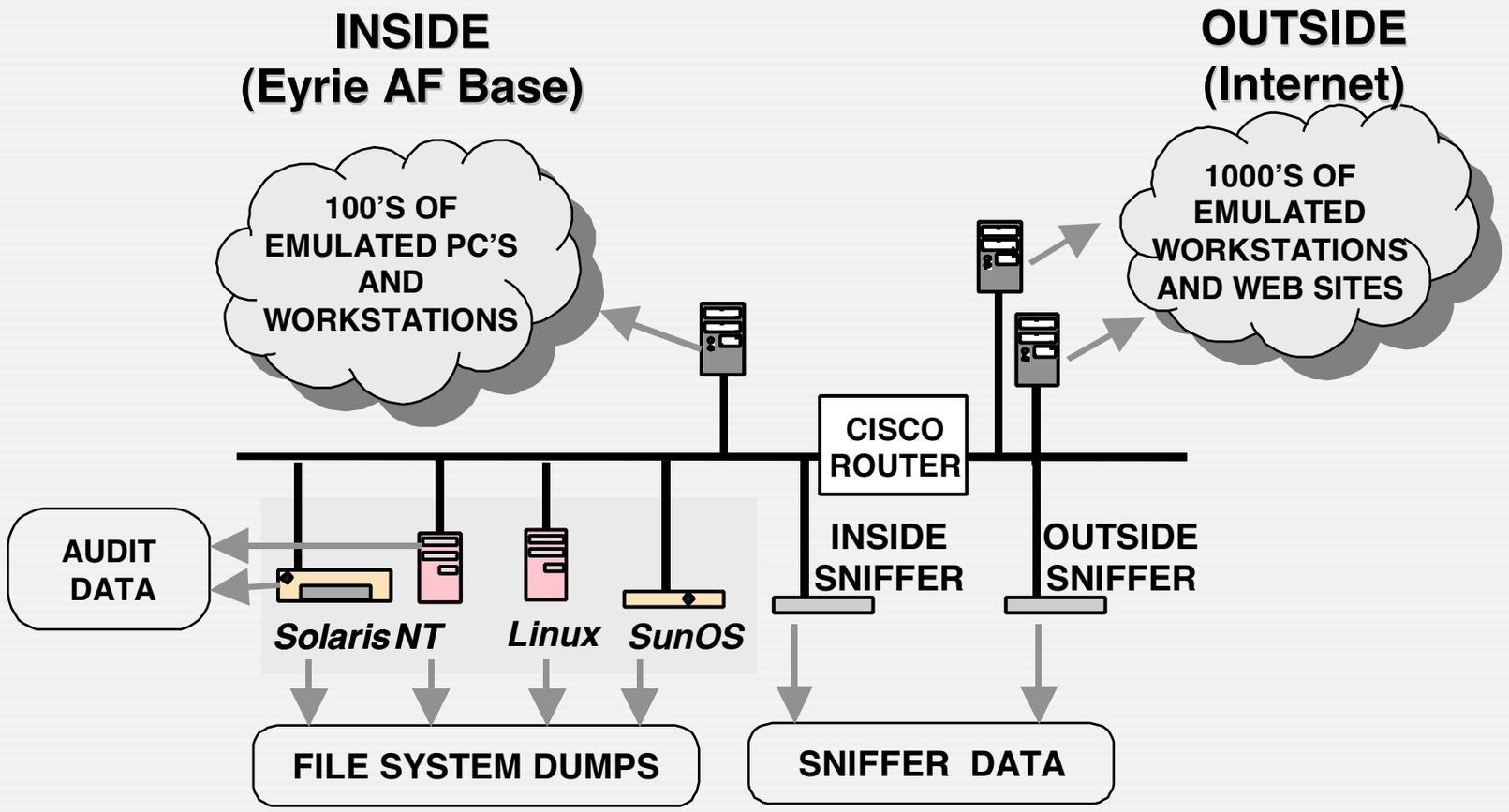


Source: Lippmann et. al. 1999

A Standard Corpus For Evaluation?

DARPA/Lincoln Labs & KDD Cup Data Sets

An attempt at providing the community with public data sets for fair NIDS evaluation



Source: Lippmann et. al. 1999

Problems

- Simulation unrealistic
- Much too regular
- Unit of analysis unclear
- Artifacts

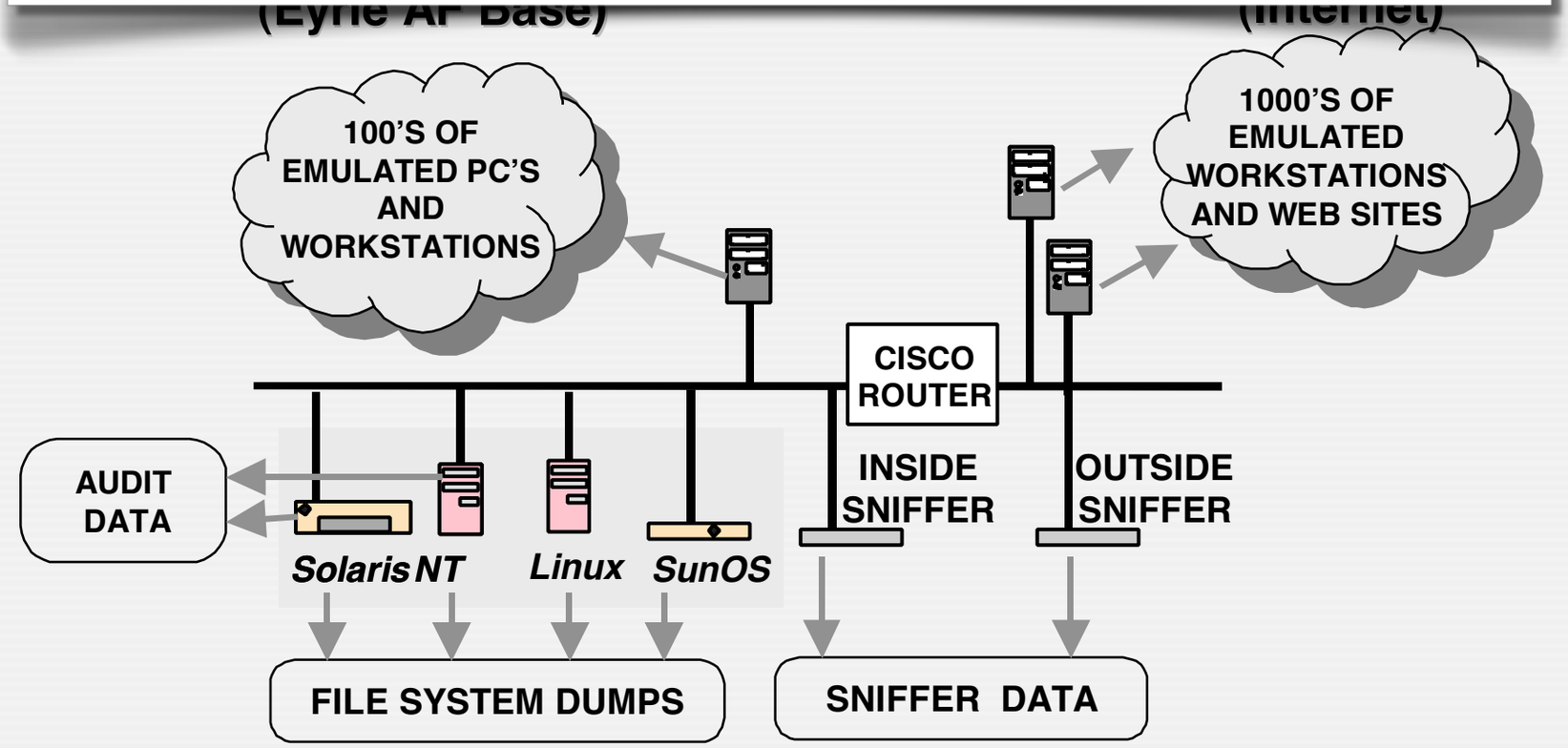
- *Now totally outdated*

A Standard Corpus For Evaluation?

DARPA/Lincoln Labs & KDD Cup Data Sets

sets for fair NIDS evaluation

Kiss Of Death for your paper!



Source: Lippmann et. al. 1999

Problems

- Simulation unrealistic
- Much too regular
- Unit of analysis unclear
- Artifacts

- *Now totally outdated*

Evaluation Data

- We need *real* network traffic for realistic evaluations.
  - The larger the environment, the better.
- Privacy/confidentiality constraints limit data sharing.
  - Operators are (understandably) reluctant to record & share network activity.
- One possible solution: scrubbing/anonymizing.
  - Trade-off between usefulness and information removed.
  - Has gained little traction because of fear that information may still leak.
- Typical approaches for sound evaluations:
  - Work with operators to get access to real network traffic, but don't share.
  - Mediated settings ("send a script", "send a student").
- But in any case: results will differ elsewhere. Always.
  - Need to clearly state setting, assumptions, and limitations.

Training Data

- A machine-learning algorithm needs to be trained.
  - With data of the *target* environment.
- Training data needs to be either *labeled* or *attack-free*.
  - Well, we are looking for novel attacks ...
  - And there's also all this background noise.
- Approaches:
  - Simulate the traffic learned from ⇒ Unrealistic.
  - Remove known attacks from real traffic ⇒ Only as good as our knowledge.
  - Filter real traffic for known good ⇒ Removes diversity.
  - Assume real traffic is attack-free ⇒ Hard to predict effect.
- Unsolved in the general case
  - In specific cases, one of the above *may* work.

Adversarial Setting

- Attackers attempt to evade detection.
  - "Flying under the radar".
  - Arms-race between attacker and defender.
- Different types of evasion:
  - Leverage specifics of the data analysis to mislead detector.
  - Mimicry attacks: pretend to be normal.
  - Gradually teach the detector to accept attacks as normal.
- Separates security most clearly from other domains.
  - Very stimulating from a theoretical perspective.
  - However, not that relevant in most practical settings.

Why is Anomaly Detection Hard?

The intrusion detection domain faces challenges that make it fundamentally different from other fields.

Outlier detection and the high costs of errors
Interpretation of results
Evaluation
Training data
Evasion risk

Why is Anomaly Detection Hard?

The intrusion detection domain faces challenges that make it fundamentally different from other fields.

Can we still make it work? Yes, by:

- Limiting the scope
- Gaining insight into the detector's capabilities

Can We Still Make it Work?

Limiting the Scope

- What attacks is the system to find?
  - The more crisply this can be defined, the better the detector can work.
  - Must include a consideration of threat model (environment; costs; exposure).
- Define a concrete task upfront, e.g.:
  - Denial-of-service floods.
  - Unauthorized code execution.
  - CGI exploits.
  - Don't go for the obvious ones ...
- Define the problem so that ML makes less mistakes:
  - Build a real classification problem.
  - Reduce variability in what's normal.
  - Look for variations of *known* attacks.
  - Use machine-learning as one tool among others.

Reducing Variability

Reducing Variability

- Select features that are more stable than others.
  - Ports hosts *accept* connections on.
  - Mapping IP to MAC addresses (in some environments)

Reducing Variability

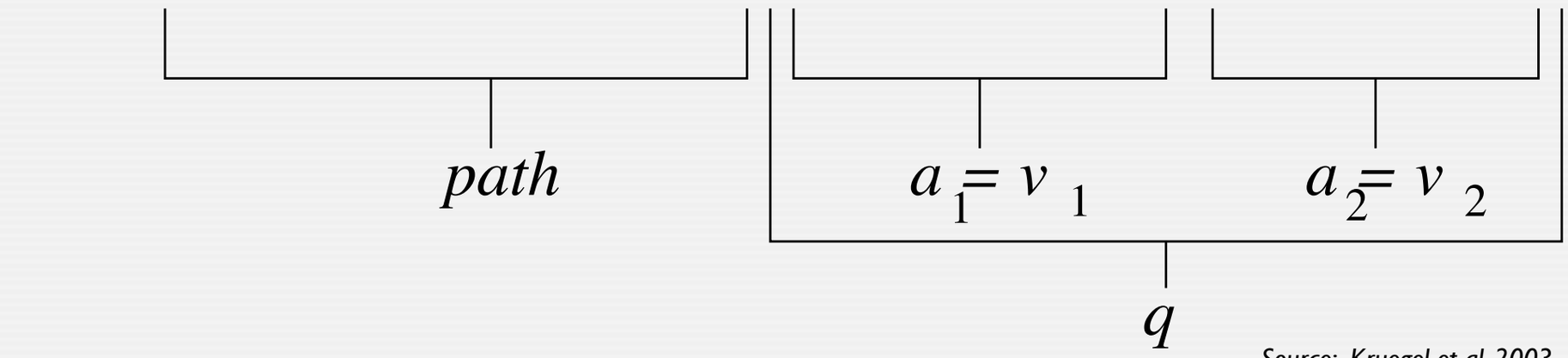
- Select features that are more stable than others.
  - Ports hosts *accept* connections on.
  - Mapping IP to MAC addresses (in some environments)
- Select features with crisp semantics.
  - Often, only the application-layer provides the right context.

Reducing Variability

- Select features that are more stable than others.
  - Ports hosts *accept* connections on.
  - Mapping IP to MAC addresses (in some environments)
- Select features with crisp semantics.
  - Often, only the application-layer provides the right context.
- Aggregation often yields stability.
  - ... by time: removes short-term fluctuations.
  - .... by subject: removes heterogeneity.
  - This is the basis for some good commercial anomaly detectors.

Focussing On A Specific Problem

Anomaly Detection of Web-based Attacks



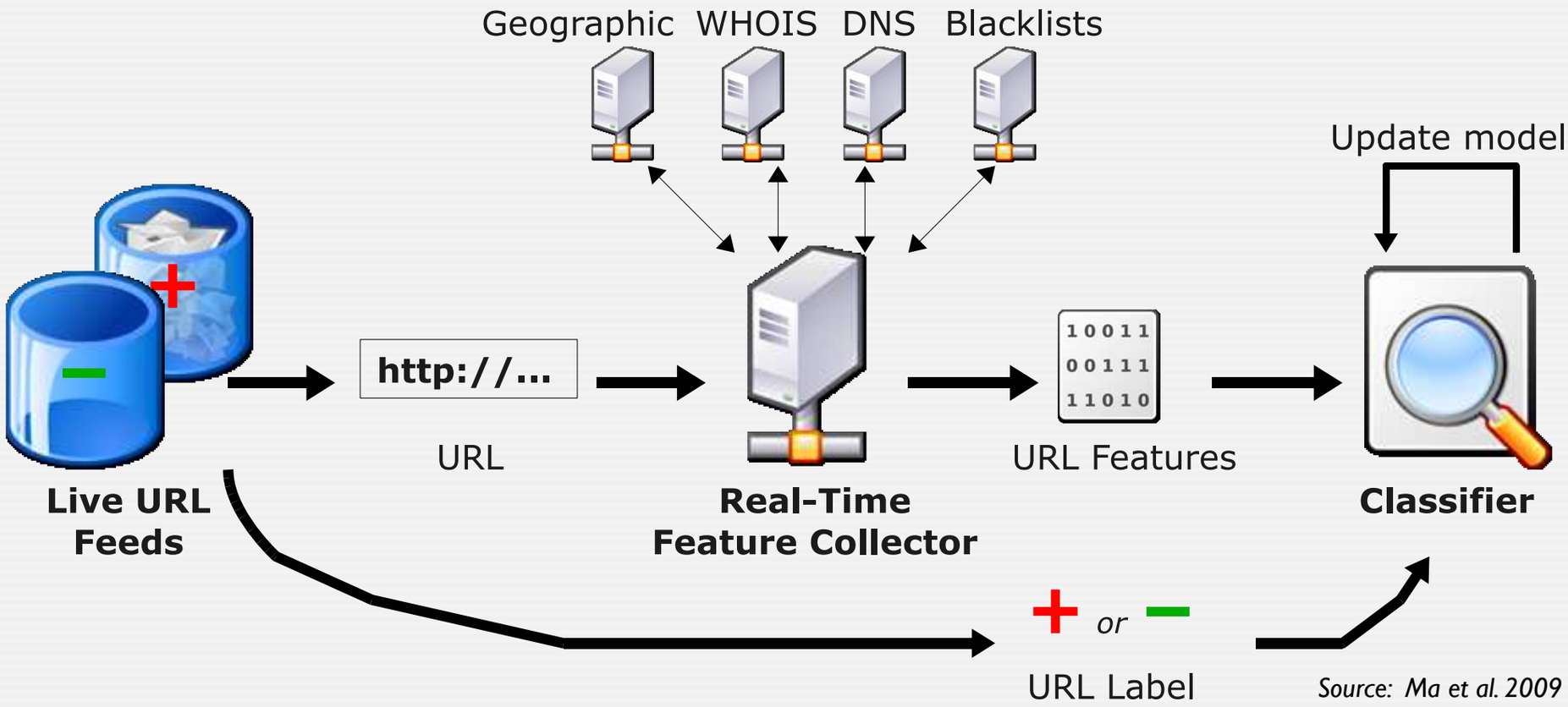
Source: Kruegel et al. 2003

Attribute Models

- Length
- Character distribution
- Grammatical structure
- Tokens
- Presence/Absence
- Order of attributes

Focussing On A Specific Problem

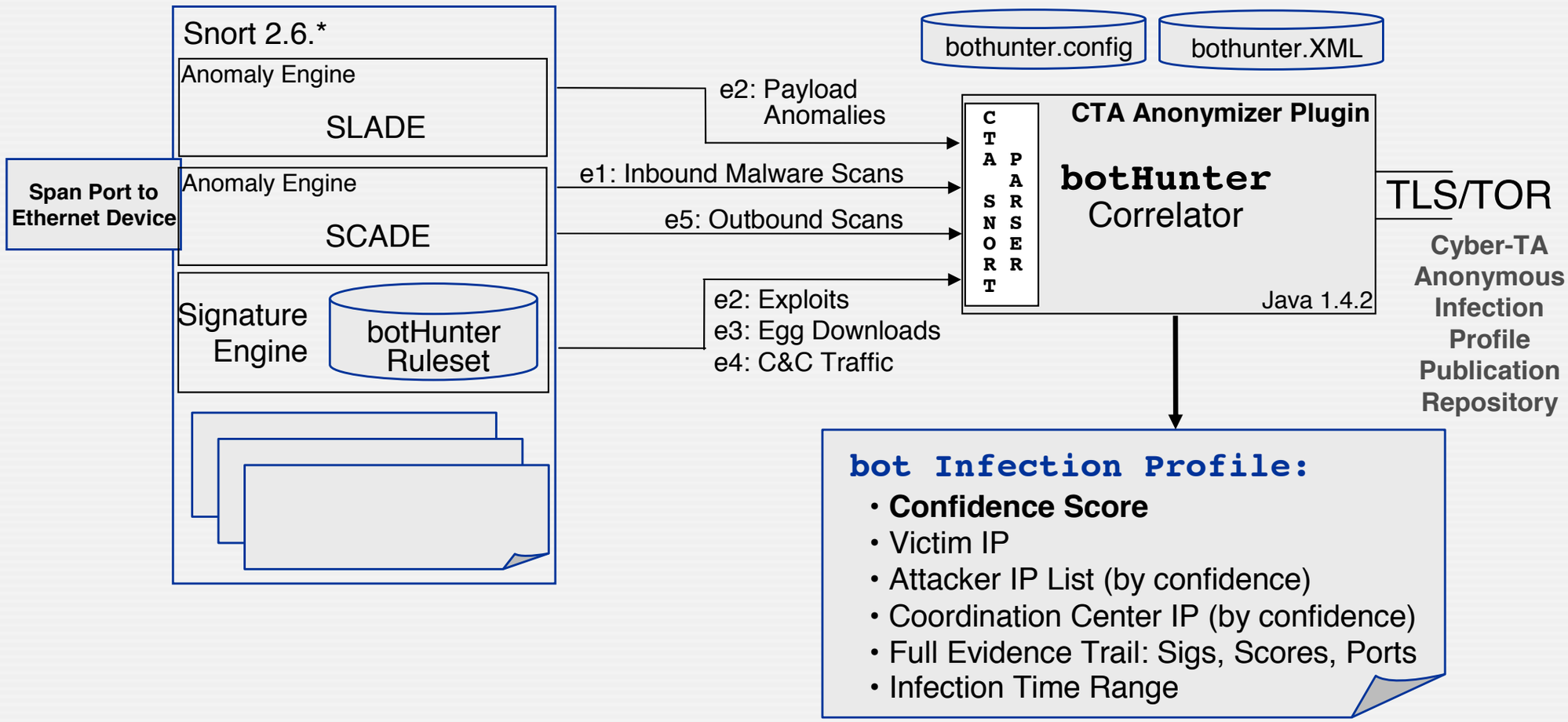
Identifying Suspicious URIs



Source: Ma et al. 2009

Machine-Learning As *One* Tool

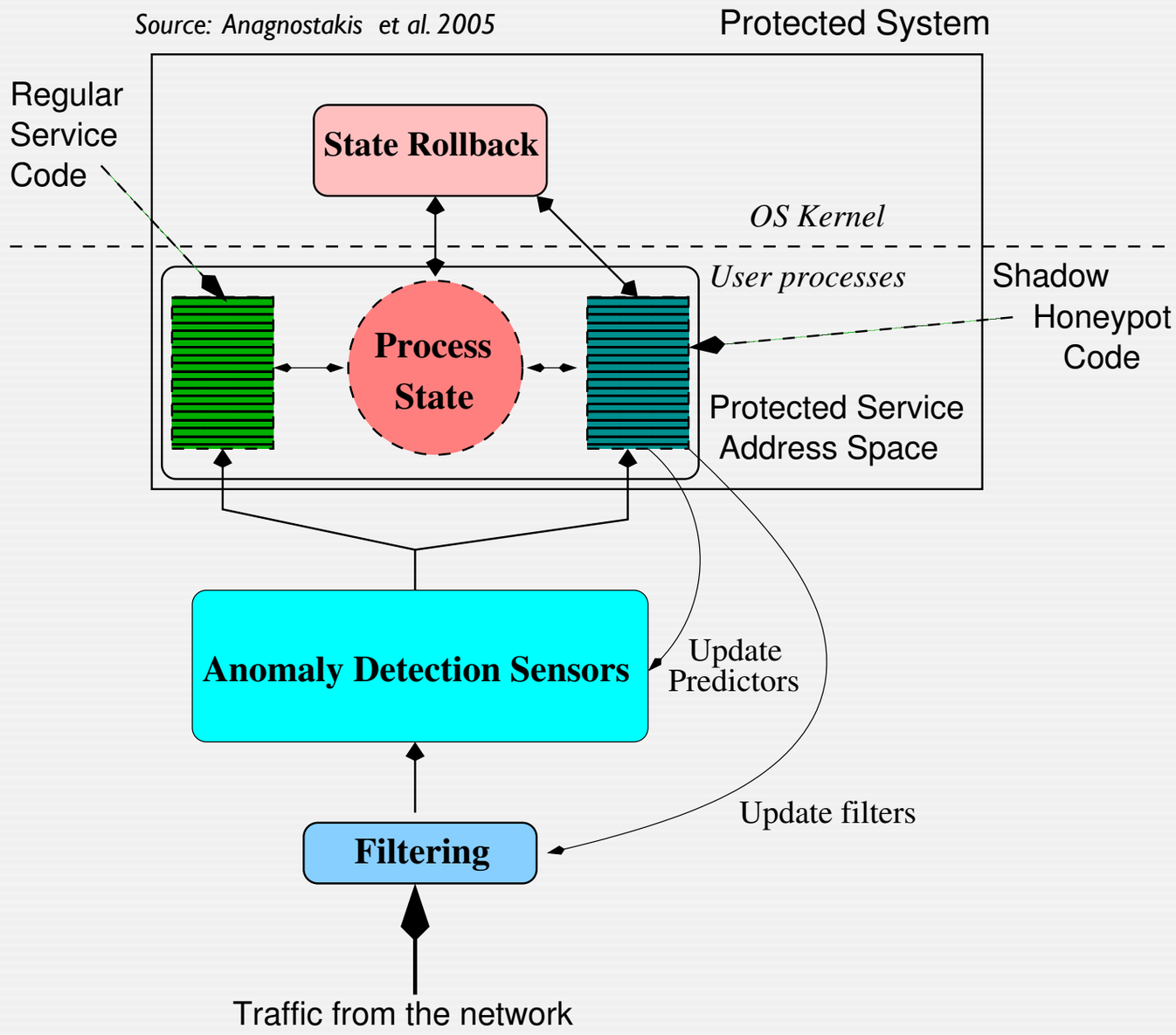
SRI's BotHunter



Source: Gu et al. 2007

Machine-Learning As *One* Tool

Shadow Honeypots



Gaining Insight

- A thorough evaluation requires more than ROC curves.
  - It's not a contribution to be slightly better than anybody else on a specific data set.
- Questions to answer:
  - What exactly does it detect *and why?*
  - What exactly does it not detect *and why not?*
  - When exactly does it break? (Evasion, performance). ("Why 6?" Tan/Maxion 2001)
- Acknowledge shortcomings.
  - We are using heuristics, that's ok. But understand the impact.
- Examine false positives and negatives carefully.
  - Needs ground-truth, and we should think about that early on.
- Examine true positives and negatives as well.
  - They tell us *how* the detector is working.

Image Analysis with Neural Networks

Tank

Image Analysis with Neural Networks

Tank

No Tank

Image Analysis with Neural Networks

Tank

No Tank

Image Analysis with Neural Networks

Tank

No Tank

Image Analysis with Neural Networks

Ta ank

Bridge the Gap

- Assume the perspective of a network operator
  - How does the detector help with *operations*?
  - With an anomaly reported, what should the operator do?
  - How can local policy specifics be included?
- Gold standard: work *with* the operators
  - If they deem the detector useful in daily operations, you got it right.
  - Costs time and effort on both sides however.

Once You Have Done All This ....

... you might notice that you now know enough about the activity you're looking for that you *don't need any machine-learning*.

- ML can be a tool for illuminating the problem space.
- Identify which features contribute most to outcome.
- ... to then perhaps build a non-machine learning detector.

Conclusion

Summary

- Approaches to network intrusion detection.
  - Host-based vs. network-based detection.
  - Misused detection and anomaly detection.
- Why is anomaly detection so hard?
  - Outlier detection and the high costs of errors.
  - Interpretation of results.
  - Evaluation.
  - Training data.
  - Evasion risk.
- Use care with machine-learning:
  - *Limit the scope* of the problem.
  - *Gain insight* into what the system does.

Conclusion

- Wanted to give a feel for *intricacies* when using *machine-learning* in the security domain.
- Bottom-line: reasonable and possible, but *needs care*.
- If you're doing anomaly detection, *understand* and *explain* what you're doing.
- If somebody hands you an anomaly detection system, *ask questions*.

Conclusion

- Wanted to give a feel for *intricacies* when using *machine-learning* in the security domain.
- Bottom-line: reasonable and possible, but *needs care*.

Open questions

"Soundness of Approach: *Does the approach actually detection intrusions? Is it possible to distinguish anomalies related to intrusions from those related to other factors?*"

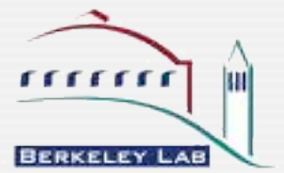
-Denning, 1997

Thanks for your attention.

Robin Sommer

International Computer Science Institute, &
Lawrence Berkeley National Laboratory

robin@icsi.berkeley.edu
http://www.icir.org



Thanks for your attention.

Robin Sommer

International Computer Science Institute, &
Lawrence Berkeley National Laboratory

robin@icsi.berkeley.edu
http://www.icir.org

... and don't use the DARPA data set!